

DAA/LANGLEY

(NASA-CR-180299) SOLUTIONS OF SOME PROBLEMS N87-26582
IN APPLIED MATHEMATICS USING PACSYNA Final
Report, Aug. 1983 - Dec. 1986 (Hampton
Inst.) 77 p Avail: NTIS HC A05/MF A01 Unclass
CSCL 12A G3/64 0063682

FINAL REPORT
OF
NASA MACSYMA PROJECT

IN-64
63682
77P.

Title of the Grant: Solutions of Some Problems
in Applied Mathematics Using MACSYMA

Grant Number: NASA Research Grant NAG-1-415

Duration: August 1983 to December 1986

by

Alkesh Punjabi and Maria Lam

Hampton University

Hampton, VA 23668

January 1987

CONTENTS

	Page
Abstract	ii
I. Introduction	1
II. Evaluation of Some Symbolic Manipulation Software (by Dr. Shah)	3
III. Representations of Shape (by Dr. Lam)	9
IV. Kinematics of Robot Arm (by Dr. Punjabi)	50
V. Contributed Papers	72

ABSTRACT

Various Symbolic Manipulation Programs have been tested to check the functioning of their commands and suitability under various operating systems. Support services for SMP have been found to be relatively better than the one for MACSYMA. The graphics facilities for MACSYMA does not work as expected under UNIX operating system. Not all the commands for MACSYMA function as described in manuals.

Shape representation has been a central issue in computer graphics and computer-aided design. Aside from appearance, there are other application dependent, desirable properties like continuity to certain order, symmetry, axis-independence and variation-diminishing properities. Several shape representations are studied, which include the Osculatory Method, a Piecewise Cubic Polynomial Method using two different slope estimates, Piecewise Cubic Hermite Form, a method by Harry McLaughlin, and a Piecewise Bezier Method. They are applied to collected physical and chemical data. Relative merits and demerits of these methods and examined.

Kinematics of a single link, non-dissipative robot arm is studied using MACSYMA. Lagranian is set-up and Lagrange's equations are derived. From there, Hamiltonian equations of motion are obtained. Equations suggest that bifurcation of solutions can occur, depending upon the value of a single parameter. Using the characteristic function W , Hamilton-Jacobi equation is derived. It is shown that H-J equation can be solved in closed form. Analytical solutions to H-J equation are obtained.

I. INTRODUCTION

This project started in August, 1983 and ended in December, 1986. From August, 1983 till August, 1986; Dr. Shantilal Shah was the Principal Investigator. Dr. Shah left Hampton University in September, 1986. Dr. Alkesh Punjabi was Principal Investigator from September, 1986 to December, 1986. Dr. John Wiggs, Dr. Maria Lam and Dr. Alkesh Punjabi have been Associate Investigators in this project for varying durations. Dr. John Wiggs is no more with Hampton University.

Hampton University has benefited by this project in a number of ways. A number of students learnt how to use symbolic manipulation programs like MACSYMA, SMP, NUMATH and others. These students were black minority students. The working knowledge and experience of symbolic manipulation programs and their application to problems will definitely help these students when they enter the market place for job.

The faculty members who were involved in the project have also benefited. They were able to learn and use one of the most important tools in modern mathematical and physical sciences research. It is generally accepted that symbolic manipulation programs are such a tool. This project helped the faculty in carrying out research using MACSYMA and SMP.

This project has also been helpful to the recently started graduate program in Applied Mathematics by way of the graduate research assistantships made available to graduate students in this program. The work done under this project may be the dissertation material for one candidate for M.S. in Applied Mathematics. The computer, the hardware and software made available to Hampton University under this project has been of immense value to the University and especially to the Computer Science

Department. These computational capabilities have facilitated the teaching of a number of courses in the computer science curricula. It has also helped the research activities of mathematics, computer science, and physics departments.

In short, one can definitely assert that this project has been of great benefit and help to the University in many ways.

On the other hand, NASA LaRC has also benefited by this project. NASA LaRC was provided access to MACSYMA and SMP on VAX computers at Hampton University. Also, under this project, the MACSYMA, SMP, and other programs were tested and the defected commands and functions which did not work or did not work as postulated were detected and pointed out. It is hoped that these aspects of the project have been of use to NASA LaRC.

Three contributed papers were presented on research done under this project. The details of these are given elsewhere in this report.

The rest of this report is organized as follows:

Sections 2, 3, and 4 detail the work or research done by Dr. Shah, Dr. Lam and Dr. Punjabi respectively under this project. Each section is written by the faculty member concerned. Section 5 gives the details of the contributed papers.

II. Evaluation of Some Symbolic Manipulation Software

In 1984, Hampton University purchased VAX-11/780 computer with 4 megabytes memory with VMS operating system and installed REX-MACSYMA software purchased from Symbolics Inc. MACSYMA is a symbolic manipulation software which was developed at M.I.T. over the period of the last 20 years. It was developed and resided on Digital Equipment Corporation PDP-10 Computer. Many scientists at M.I.T. had also created library packages to solve various problems in pure and applied mathematics and it was made available for public use under shared libraries under MACSYMA. MACSYMA has also capabilities for point-graphs and 2 and 3 dimensional line-graphs. MACSYMA software became an important tool in research and development in solving difficult problems in the fields of pure and applied mathematics, engineering, physics, aerodynamics and several other disciplines. In 1983, M.I.T. sold the marketing and support of MACSYMA to SYMBOLICS Inc.

SYMBOLICS Inc. started distribution of MACSYMA software which ran on VAX-11/780 computer under VMS operating system in 1983. When it was installed on VAX-11/780 computer running under VMS at Hampton University in 1984, it was able to support most of MACSYMA commands available in the language. The investigators spent spring and summer of 1984 in beta testing of the software. The line-graphic support was not available on REX-MACSYMA. There were some other bugs in few MACSYMA commands. The shared library packages were available on REX-MACSYMA software for use, but none of them were working. This information was conveyed to Symbolics Inc. at the MACSYMA Conference at Schenectady, N.Y. in summer of 1984. People from Symbolics Inc. told the investigators that shared libraries were not the part of MACSYMA software package, they were not responsible for its support,

and Hampton University was not supposed to have it on its machine. The Symbolics Inc. also acknowledged that they had problems in supporting of line-graphics, both two and three dimensions, capabilities of MACSYMA at the time and they were working on the problems. The execution speed of various commands of REX-MACSYMA was also considerably slower compared to the software running at M.I.T.

In spring of 1984, Wollongong Inc. came out with a software named EUNICE, which works like UNIX operating system. It supports all of the UNIX operating system commands and runs under VMS operating system. When it is installed on VAX-11 machine running under VMS operating system, the user has advantage of using both VMS and UNIX operating system commands. EUNICE software imulates UNIX operating system running under VMS. Investigators inquired with Symbolic Inc. whether they had developed a version of MACSYMA which runs under EUNICE operating systems and whether EUNICE-MACSYMA will support line-graphics capabilities of MACSYMA. Symbolics Inc. informed the investigators that they had EUNICE-MACSYMA and the version supported line graphics. In 1985, Hampton University investigators purchased and installed EUNICE operating system from Wollongong Group Inc., and swapped REX-MACSYMA for EUNICE-MACSYMA with Symbolics Inc. Investigators also purchased SMP software developed by Inference Corporation. SMP is a symbolic manipulation program like MACSYMA, which has capabilities to perform symbolic computations and graphics for two and three dimensions. SMP software can run under either VMS operating system or UNIX operating system on VAX-11 computers. Investigators have performed comparison of execution times of SMP commands and MACSYMA commands running under VMS operating system. SMP software supports graphics capabilities on various graphics terminals. EUNICE-MACSYMA, like REX-MACSYMA can do point plots, but does not support

line graphics either on Digital's GIGI graphics terminals or Tektronix graphics terminals. This was again brought to the attention of people at Symbolics Inc. Inquiries were again made with Symbolics Inc. whether the UNIX version of MACSYMA available from Symbolics Inc. was able to support line-graphics and whether they can provide the names of the institutions or companies which are currently users of UNIX-MACSYMA. The reply of Symbolics Inc. was that UNIX-MACSYMA does support line graphics on Tektronix terminals and due to some Symbolic Inc. policy, they cannot give out the names of UNIX-MACSYMA users. Hampton University installed UNIX operating system on its VAX-11/780 computer in February of 1986 and also installed UNIX version of MACSYMA on VAX-11/780 computer. Experiments were performed to test graphics capabilities of UNIX-MACSYMA on tektronix terminals. The results of these experiments showed that, the software was not working correctly as desired. Some of the drivers routines required to support graphics were not available. Symbolics Inc. tried to evade the responsibilities for not having driver routines instead they tried to find fault with UNIX operating system and tektronix graphic terminals. Finally in June of 1986, Jeffery P. Golden at Symbolics Inc., looked into the problem and found out that graphics capabilities on UNIX-MACSYMA were not working properly. In August 1986, at the Stanford University meeting, he told that he and other people at Symbolics Inc. were working on the problem and it will be sometime before the problem is corrected.

Since Symbolics Inc. acquired the distribution of MACSYMA software, their support for the software is totally unsatisfactory and the graphic capabilities (line) is not working on any of their versions REX, EUNICE or UNIX. So far, they are not truthful about giving information on graphics capabilities of MACSYMA. They are charging about \$350.00 for support

service on MACSYMA, and customers do not get any support from the service. The shared libraries are not available for MACSYMA users.

SMP software is comparatively new and young in the market. Some of its commands are also not working properly. But their support is very excellent. The graphics capabilities on SMP is very good and works correctly. Some of the commands like 'XTHRU' which is available in MACSYMA are not implemented on SMP. But users can get the required results by implementing those with existing commands. The user's manual for SMP is not easy to understand for implementing and using all the capabilities of SMP. The people at Inference Corporation are willing to help and guide the users to overcome the shortcomings of the manual. The investigators found that one of the routines in SMP library, namely FELIP had some bug in it and it was reported to Inference Corporation. We believe, it is corrected now. Richard Fateman of the University of California at Berkeley, performed various experiments with SMP and published various SMP shortcomings of the software in ACM's SIGSAM BULLETIN of August 1985. These experiments were performed on SMP version 1.3 Inference Corporation updated version 1.5 of SMP still has some of the old problems. For example, it is unable to factor simple algebraic expression $X^6 - a^6$. It exhausts memory even if you start fresh. This is not an isolated case. Inference claims that these problems are scheduled for work prior to next release.

The other symbolic softwares like REDUCE and MAPLE is also available for VAX-11/780 computers. MAPLE and REDUCE, both do not support graphics. MAPLE is good for the university environment, in the sense, it does not take up lot of memory, and many users can access the same time. One MACSYMA user requires 1.5 megabyte compared to 100K for MAPLE. MAPLE is still under development and it has limited capabilities compared to MACSYMA. REDUCE is

around for a long time and it is widely used in industries and universities in U.S. and Europe. Recently, it was announced that REDUCE is now available for CDC-CYBER-170 series.

Of all different kinds of symbolic manipulation software, according to these investigators, MACSYMA has the best capabilities. But the support needed for maintenance and development for users of MACSYMA is not available from Symbolics Inc. It seems they are more interested in implementing MACSYMA on single user machines, rather than supporting MACSYMA software for multiuser machines like VAX-11 series. In these respects, SMP software is much desirable due to its capabilities and support given by Inference Corporation.

The investigators have performed comparison of execution times of some commands in MACSYMA and SMP.

GENTRAN:

A software package implemented in RLISP is now available for code generation and translation and it runs under REDUCE. It generates complete numerical programs directly from REDUCE by transforming REDUCE prefix forms into formatted FORTRAN, RATFOR or C code. Assignment statements, control structures, type declarations and subprogram headings can be generated from algorithmic specifications and symbolically derived formulas. An expression segmentation facility breaks large expressions into subexpressions of manageable size, and an interface to an existing code optimizer allows sequences of assignment statements to be replaced by their optimized equivalents. In addition, special file-handling facilities allow code generation to be guided by template files and output to be redirected to one or more files. GENTRAN provides the flexibility necessary to handle most

code generation application under REDUCE. Wm Leler and Neil Soiffer, of Tektronix Inc. has developed a system which is available for workstations to do graphics and pretty pointing for REDUCE. REDUCE was chosen over other symbolic algebra systems primarily because of its low cost and availability of source code.

III. REPRESENTATIONS OF SHAPE

Introduction:

In computer graphics and computer-aided design, the problem of shape representation and design is a central issue. We are not only concerned with pictures but also with providing a geometrical model or database which is used for design, analysis and production. Shape representation raises many difficulties. There are no well-established criteria for arriving at an appropriate mode of description. Approaches to shape and form description are usually highly intuitive and adhoc responses to practical needs.

In the early sixties, Charles Coons of M.I.T. initiated some elegant and precise methods to represent shapes. His methods were quickly adopted by industry; generalized and improved by many researchers. Coons employed parametric forms to describe curves and surfaces to obtain axis-independent shape descriptions. This also allows the description of closed curves. His methods are also piecewise. No attempt is made to represent an entire curve or surface by a single analytic function. Curves and surfaces are constructed piecewise, then they are sewn together with specified continuity condition. P. Bezier proposed a similar idea. Coons' blending functions become the basis functions for Hermite interpolation and Bezier's blending functions are the basis functions for Bernstein approximation.

A mathematical representation of shape can be obtained in two basic ways: design and fitting. In design, the problem is to create (or modify an existing) shape which satisfies certain constraints. It usually involves making interactive changes in shapes and displaying them in real-time. In fitting, the problem is to obtain a mathematical representation for an

existing shape which is not mathematically defined but exists as a physical model from which data points can be measured.

Traditionally, there are two approaches to these problems: approximation or interpolation. Interpolation means the shape matches the given data exactly and approximation means one nearly matches the data. Most scientific representation of informations requires approximations at some level. The approximation might occur at the level of equations that model the physical phenomenon, or at the level of the numerical solution of these equations.

Properties of Shape Representation:

The requirements of the application and of the computer used to generate these shapes (curves and surfaces) suggest that their representation should have certain properties. Some of the important requirements are listed below.

1. The representation should allow multiple values as in the case of closed curves or surfaces.
2. Some smoothness is desired. It might be C^0 , C^1 or C^2 continuity.
3. The representation must not lend itself to too much computation and computer memory requirement.
4. The shape of an object is independent of its orientation with respect to arbitrarily chosen coordinate axes. Hence it is desirable that its mathematical representation be axis independent.

5. The method to generate these shapes can be local or global. Local method is probably preferred because shape can be generated as soon as some data points are available. It is also easier to control the designed shape by making some modifications around the trouble area.
6. The mathematical representation should be variation-diminishing. It should smooth small irregularities of the data points instead of amplifying them.
7. It is desirable that the same curve is generated when the data is scanned from left-to-right or right-to-left. Hence the representation should have symmetry property.
8. The representation should be versatile so that it provides a variety of shapes.

The Representations:

A survey has been conducted on some of the commonly used shape representations¹⁻⁴. None of these methods requires derivative informations directly. Unlike what appeared in many articles on this topic (shape representation), the data used in this study are collected physical and chemical data and not artificial. SMP is used exclusively to conduct this study. Result of this study is described in the following sections. The discussion is based on some of the criteria described in the previous section. Note that Bezier method is the only approximation, the rest provides interpolation.

Assume there are N data points. They are represented as $P_1(x_1, y_1)$,

$P_2(x_2, y_2) \dots, P_N(x_N, y_N)$ where $x_1 < x_2 < \dots < x_N$.

Method A: Osculatory Method

Add two dummy data points P_0 and P_{N+1} , one before the first data point and one after the last point.

For every $i = 1, 2, \dots, N$, define a quadratic polynomial $Q_i(x) = ax^2 + bx + c$ on $[x_i, x_{i+1}]$ such that $Q_i(x)$ passes through P_{i-1}, P_i, P_{i+1} .

On $[x_i, x_{i+1}]$, blend $Q_i(x), Q_i(x), Q_{i+1}(x)$ to form $c_i(x)$ by using the two blending functions as indicated below.

$$c_i(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}} Q_i(x) + \frac{x - x_i}{x_{i+1} - x_i} Q_{i+1}(x)$$

These polynomial $[c_i(x) : i = 1, 2, \dots, N-1]$ form a piecewise smooth interpolant.

Method B: A Piecewise Cubic Polynomial Method

Add a dummy point P_0 before the first data point. Add P_{N+1} after the last point.

For every $i = 1, 2, 3, \dots, N+1$, calculate the slope m_i of line segment $\langle P_{i-1}, P_i \rangle$.

$$m_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$$

An estimated slope s_i associated with each point $P_i(x_i, y_i)$ is then determined.

For every $i = 1, 2, \dots, N$,

$$s_i = 0.5 (m_i + m_{i+1})$$

Define a cubic polynomial function $c_i(x)$ on $[x_i, x_{i+1}]$ such that $c_i(x)$ interpolates P_{i-1}, P_i, P_{i+1} and $c'_i(x_i) = s_i$.
 Piece these polynomials $[c_i(x) : i = 1, 2, \dots, N-1]$ to form a smooth interpolant over $[x_1, x_N]$.

Method B': A Piecewise Cubic Polynomial Method with Akima's Slope Estimate

This method is identical to Method B except that the estimated slope s_i at point $P_i(x_i, y_i)$ is defined as:

$$s_1 = m_2$$

$$s_2 = 0.5 (m_2 + m_3)$$

and for every $i = 3, \dots, N-2$

$$s_i = \frac{|m_{i+2} - m_{i+1}| m_i + |m_{i-1} - m_i| m_{i+1}}{|m_{i+2} - m_{i+1}| + |m_{i-1} - m_i|}$$

which is a weighted average of m_i, m_{i+1} .

$$s_{N-1} = 0.5 (m_{N-1} + m_N)$$

$$s_N = m_N$$

Method C: A Piecewise Cubic Hermite Polynomial

The slope S_i at data point $P_i(x_i, y_i)$ is estimated as in Method B. On each interval $[x_i, x_{i+1}]$, a cubic polynomial $c_i(t)$ in parameter t is constructed by using blending functions as indicated.

$$c_i(t) = (2t^3 - 3t^2 + 1) P_i + (-2t^3 + 3t^2) P_{i+1} + (t^3 - 2t^2 + t) P'_i + (t^3 - t^2) P'_{i+1}.$$

where $0 \leq t \leq 1$.

These c_i 's , $i = 1, 2, \dots, N-1$ give a smooth interpolant.

Method D: McLaughlin's Method

Each data point with the exception of the last point is classified as "regular" or "irregular". If a point $P_i(x_i, y_i)$ is irregular, a line segment is drawn from this point to the next point $P_{i+1}(x_{i+1}, y_{i+1})$. If a point is regular, a break point $B(x', y')$ will be inserted between P_i and P_{i+1} , then a parabola passing through P_i and P_{i+1} will be drawn.

Method E: Piecewise Bezier Method

Four control points will be used to construct a Bezier curve.

$$P(t) = \sum_{k=0}^3 \binom{3}{k} t^k (1-t)^{3-k} P_{i+k}$$

where $0 \leq t \leq 1$.

Piece these curves together to get an approximation.

Data:

The data used to generate Figures 1 to 20 are given in Tables 1-5. Water samples are collected from the same station at a depth of one meter. Some physical measurements are taken. The samples are also analyzed for some of their chemical contents.

Variable x represents time in the unit of weeks, y represents the measurement of a physical or chemical variable at low tide or high tide. Table 1 shows Dissolved Silica content (in mg/l) at low tide (y_1) and high tide (y_2). Similarly, Tables 2-5 show the measurements of Salinity

Conductivity, pH, Total Kjeldahl Nitrogen and Total Phosphate in appropriate units at low tide and high tide.

x	y1	y2
1	1.4	1.3
2	1.8	0.3
5	1.7	1.6
6	1.6	1.7
7	1.6	1.5
9	0.3	0.1
11.5	0.1	0.8
14	0.2	1.8
17	0.9	2.2
22	0.4	0.8

Table 1. Dissolved Silica in mg/l

x : time in weeks
y1 : low tide data
y2 : high tide data

x	y1	y2
1	20.5	21.1
2	23.1	23.2
5	25.2	26.0
6	27.6	25.5
7	19.8	19.7
9	20.9	29.1
11.5	26.4	23.1
14	23.8	21.4
17	25.0	21.1
22	23.9	24.6

Table 2. Salinity Conductivity in mg/l

x : time in weeks
y1 : low tide data
y2 : high tide data

x	y1	y2
1	8.45	8.28
2	7.98	8.45
5	8.62	8.31
6	7.87	8.16
7	8.06	7.81
9	8.46	8.23
11.5	7.77	8.21
14	7.68	8.10
17	8.12	8.23
22	8.25	7.98

Table 3. pH in standard unit
x : time in weeks
y1 : low tide data
y2 : high tide data

x	y1	y2
1	0.351	0.909
2	0.158	0.342
5	0.698	1.173
6	0.868	0.884
7	0.765	0.571
9	0.500	0.621
11.5	1.320	1.084
14	0.552	0.705
17	1.912	1.111
22	1.088	0.533

Table 4. Total Kjeldahl Nitrogen in mg/l
x : time in weeks
y1 : low tide data
y2 : high tide data

x	y1	y2
1	0.134	0.182
2	0.059	0.157
5	0.065	0.042
6	0.052	0.053
7	0.080	0.100
9	0.042	0.060
11.5	0.190	0.192
14	0.018	0.034
17	0.157	0.171
22	0.041	0.000

Table 5. Total Phosphate in mg/l
x : time in weeks
y1 : low tide data
y2 : high tide data

Conclusion:

The first three methods (A, B, B') described in Section III define polynomials on $[x_i, x_{i+1}]$ using explicit form

$$y = f(x).$$

Hence none of them permits multiple-valued shapes. Cubic Hermite form, Bezier curve and McLaughlin's method provide curves of single-valued functions as well as multiple-valued curves.

The osculatory method (A) matches first order derivatives at the two end points of interval $[x_i, x_{i+1}]$ as well as the data points $P_i(x_i, y_i)$ and $P_{i+1}(x_{i+1}, y_{i+1})$. It is of order C^1 . This can be seen from the following direct calculation. From Section III,

$$c_i(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}} q_i(x) + \frac{x - x_i}{x_{i+1} - x_i} q_{i+1}(x) \quad \text{on } [x_i, x_{i+1}]$$

$$c_i'(x) = \frac{q_i(x)}{x_i - x_{i+1}} + \frac{x - x_{i+1}}{x_i - x_{i+1}} q_i'(x) + \frac{q_{i+1}(x)}{x_{i+1} - x_i} + \frac{x - x_i}{x_{i+1} - x_i} q_{i+1}'(x)$$

$$c_i'(x_i) = \frac{q_i(x_i)}{x_i - x_{i+1}} + q_i'(x_i) + \frac{q_{i+1}(x_i)}{x_{i+1} - x_i}$$

$$= \frac{q_i(x_i) - q_{i+1}(x_i)}{x_i - x_{i+1}} + q_i'(x_i)$$

$$= \frac{y_i - y_i}{x_i - x_{i+1}} + q_i'(x_i)$$

$$= q_i'(x_i).$$

Similarly,

$$\begin{aligned} c'_i(x_{i+1}) &= q'_{i+1}(x_{i+1}) \\ &= c'_{i+1}(x_{i+1}). \end{aligned}$$

Methods B, B' define curves over $[x_i, x_{i+1}]$ in such a way so that only P_i , P_{i+1} and the first order derivative at x_i are matched. As a result, they only provide C^0 curves. Method C generates piecewise cubic Hermite polynomials which are C^1 continuous. McLaughlin's method is C^0 continuous since derivatives are not considered directly by this method at all. Bezier method can supply continuity of any order provided the data points satisfy certain conditions. In this investigation, the data points are collected data and will not satisfy those conditions. Hence the Bezier curves are only C^0 continuous.

All methods described in Section III are local methods. Bezier method is actually a global method. But in this investigation, a piecewise Bezier curve is used instead of a single Bezier curve therefore it is considered as a local method.

Bezier curves are symmetric due to the symmetry of the Bezier blending functions. Take four consecutive data points $P_i, P_{i+1}, P_{i+2}, P_{i+3}$, they generate

$$P(t) = \sum_{k=0}^3 \binom{3}{k} t^k (1-t)^{3-k} P_{i+k}, \quad 0 \leq t \leq 1$$

The points $P_{i+3}, P_{i+2}, P_{i+1}, P_i$ in this order (from right-to-left) generate

$$R(t) = \sum_{k=0}^3 \binom{3}{k} t^k (1-t)^{3-k} P_{i+3-k} \quad 0 \leq t \leq 1$$

Take any t_0 such that $0 \leq t_0 \leq 1$

$$P(t_0) = \sum_{k=0}^3 \binom{3}{k} t_0^k (1-t_0)^{3-k} P_{i+k}$$

$$R(1-t_0) = \sum_{k=0}^3 \binom{3}{k} (1-t_0)^k t_0^{3-k} P_{i+3-k}$$

Let $j = 3-k$, then

$$\begin{aligned} R(1-t_0) &= \sum_{j=3}^0 \binom{3}{3-j} (1-t_0)^{3-j} t_0^j P_{i+j} \\ &= \sum_{j=3}^0 \binom{3}{j} t_0^j (1-t_0)^{3-j} P_{i+j} \\ &= P(t_0). \end{aligned}$$

McLaughlin's method is not symmetric. The determination of the break point B depends on some "previous" points. While moving from right-to-left along the x -axis, these "previous" points are not the same as those associated with moving from left-to-right. In general, different parabolas will be constructed between P_i and P_{i+1} . However, the shape of the generated curve is consistent in the sense that if P_i is determined as regular and a parabola is found between P_i and P_{i+1} , when traversing from right-to-left, P_{i+1} will be determined as regular and a parabola (a different one) will be constructed between P_{i+1} and P_i .

Cubic Hermite form is also symmetric. From P_i to P_{i+1} , one has curve

represented as:

$$P(t) = (2t^3 - 3t^2 + 1) P(0) + (-2t^3 + 3t^2) P(1) + (t^3 - 2t^2 + t) P'(0) + (t^3 - t^2) P'(1) \quad 0 \leq t \leq 1.$$

From P_{i+1} to P_i , one will have

$$R(t) = (2t^3 - 3t^2 + 1) P(1) + (-2t^3 + 3t^2) P(0) + (t^3 - 2t^2 + t) (-P'(1)) + (t^3 - t^2) (-P'(0)) \quad 0 \leq t \leq 1.$$

For every t in $[0,1]$

$$\begin{aligned} R(1-t) &= [2(1-t)^3 - 3(1-t)^2 + 1] P(1) + [-2(1-t)^3 + 3(1-t)^2] \cdot \\ &\quad P(0) + [(1-t)^3 - 2(1-t)^2 + (1-t)] (-P'(1)) + \\ &\quad [(1-t)^3 - (1-t)^2] (-P'(0)) \\ &= (2 - 6t + 6t^2 - 2t^3 - 3 + 6t - 3t^2 + 1) P(1) + \\ &\quad (-2 + 6t - 6t^2 + 2t^3 + 3 - 6t + 3t^2) P(0) + \\ &\quad (1 - 3t + 3t^2 - t^3 - 2 + 4t - 2t^2 + 1 - t) (-P'(1)) + \\ &\quad (1 - 3t + 3t^2 - t^3 - 1 + 2t - t^2) (-P'(0)) \\ &= (-2t^3 + 3t^2) P(1) + (2t^3 - 3t^2 + 1) P(0) + \\ &\quad (t^3 - t^2) P'(1) + (t^3 - 2t^2 + t) P'(0) \\ &= P(t). \end{aligned}$$

The remaining methods do not have the symmetry property.

A Bezier curve is independent of the coordinate system used to measure the locations of control points. The same is true for the McLaughlin's method because both use parametric form and the methods involve only control points (and also beak point for the latter). Cubic Hermite form is not axis independent despite parametric form is also used. It depends on first order derivatives as well as data points. These first order derivatives do not undergo the same transformation as the axes. The other methods are also axis dependent.

The piecewise Bezier method is only C^0 . Curves generated by this

method in general contain corners and/or cusps. Nonetheless the method provides very interesting curves. As can be expected, McLaughlin's method generates too many line segments along the process. Some curves are made up of line segments only. However, when some line segments are mixed with a few parabolas, the results are quite interesting and pleasing to the eyes. These two methods could be valuable to those applications that call for corners or cusps. The cubic Hermite method does not produce satisfactory results. The curves have too many twists and turns. Method A gives the smoothest curves. On a few occasions, the curve does overshoot data points. Method B, B' give very similar results. Curves are relatively smooth, though not as smooth as those of A. They also produce some corners. The overshoot parts are not as noticeable as those of A. Some curves produced by B' appear to be a bit tighter in some areas.

The approximate amount of computer time in the unit of seconds needed to generate the curves of Figures 1-20 are given in the following table. These curves are generated by SMP version 1.5.0 running on a VAX 11/780 and a DEC GIGI is used as graphic display.

METHOD

	A	B	B'	C	D	E
DS	12.03	11.15	11.95	33.12	22.35	22.88
SC	12.05	10.98	11.78	33.12	15.83	22.77
pH	12.10	11.08	11.71	33.13	18.35	22.85
TKN	12.00	10.97	11.65	33.20	19.85	22.73
TP04	11.83	10.73	11.51	33.25	13.27	22.75
DS	11.97	11.07	11.68	33.33	25.20	22.68
SC	12.05	11.02	11.68	33.15	26.57	22.90
pH	11.90	11.28	11.57	33.22	22.78	22.90
TKN	11.91	10.98	11.62	33.07	18.43	22.65
TP04	11.97	10.80	11.70	33.33	15.85	22.02

Table 6. Approximate time needed in sec to generate curve

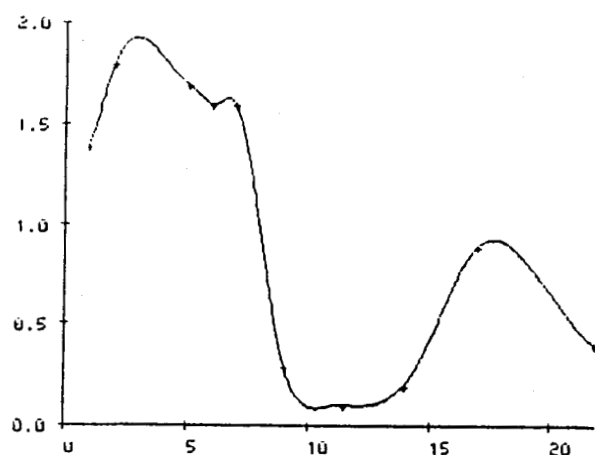
Abbreviations: DS - Dissolved Silica
 SC - Salinity Conductivity
 TKN - Total Kjeldahl Nitrogen
 TP04 - Total Phosphate
 l - low tide
 h - high tide

References:

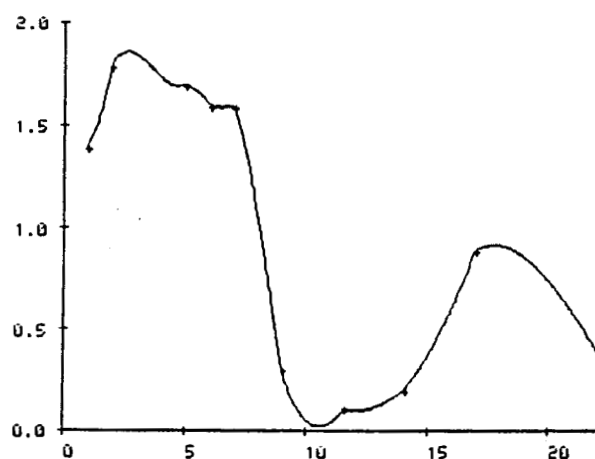
1. Brodlie, K.W., Mathematical Methods in Computer Graphics and design, Academic Press, Orlando, Florida, 1980.
2. McLaughlin, H.W., Shape-Preserving Planar Interpolation : An Algorithm, IEEE Comp. Graphics and Appl., Vol. 3, 1983, pp. 58-67.
3. Foley, J.D. and Van Dam, A., Fundamentals of Interactive Computer Graphics, Addison Wesley, Reading, Massachusetts, 1984.
4. Forrest, A.R., On Coons and Other Methods for the Representation of Curved Surfaces, Computer Graphics and Image Processing, Vol. 1, No. 4, 1972, pp. 341-359.

Figures:

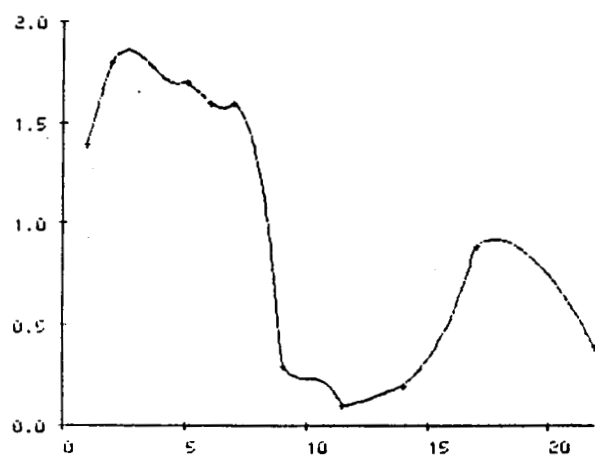
Dissolved Silica (low tide, DRAW1)



Dissolved Silica (low tide, DRAW2)



Dissolved Silica (low tide, DRAW3)



Dissolved Silica (low tide, DRAW4)

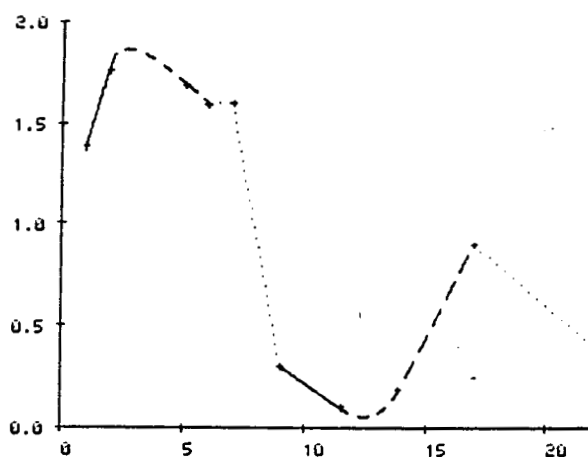
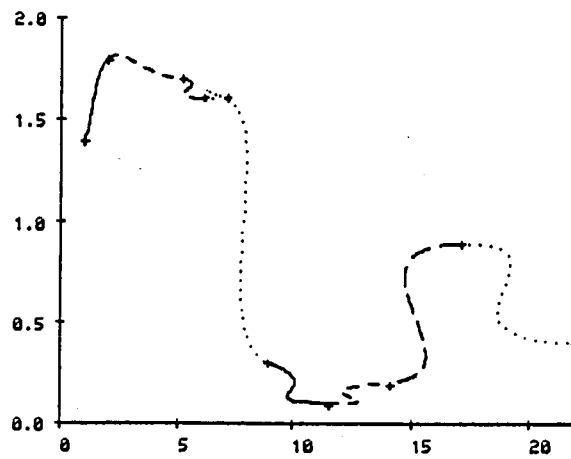


Figure 1. Dissolved Silica at low tide

DRAW 1 - Method A
 DRAW 2 - Method B
 DRAW 3 - Method B'
 DRAW 4 - Method D

Dissolved Silica (low tide, Hermitel)



Dissolved Silica (low tide, Bezier)

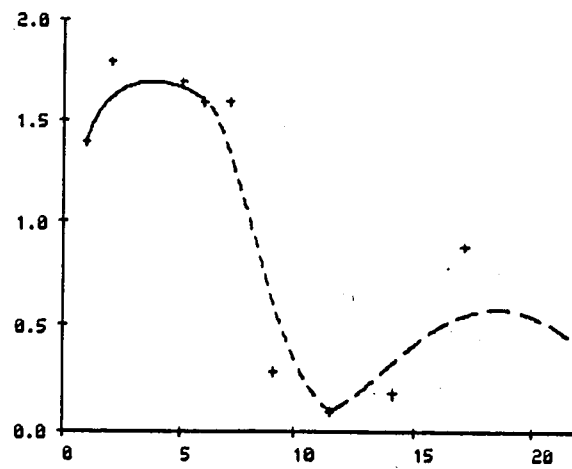
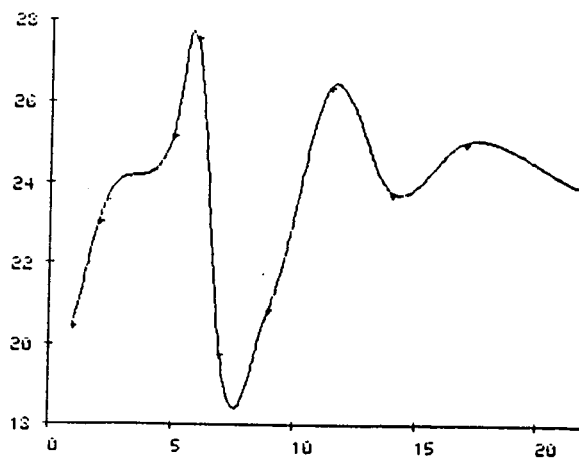
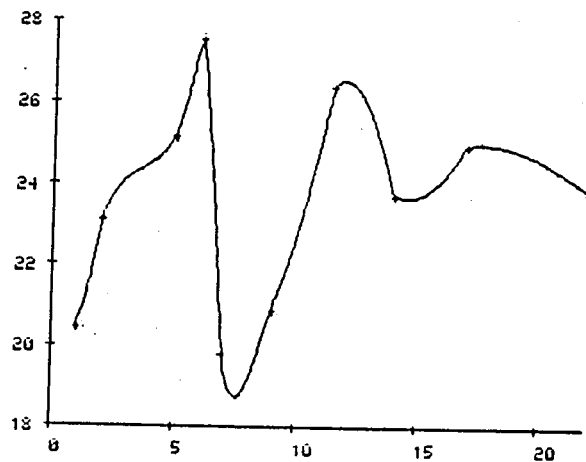


Figure 2. Dissolved Silica at low tide

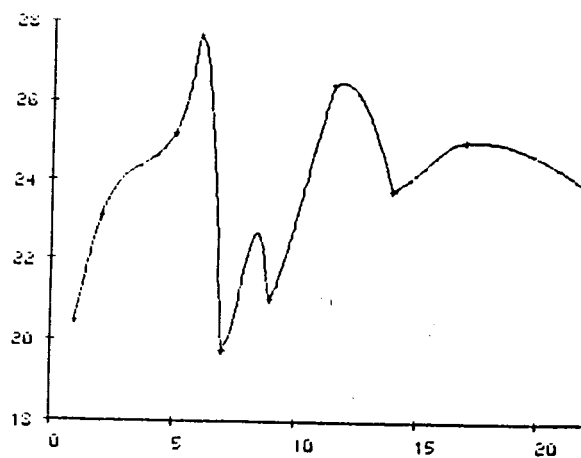
Salinity Conductivity (low tide, DRAW1)



Salinity Conductivity (low tide, DRAW2)



Salinity Conductivity (low tide, DRAW3)



Salinity Conductivity (low tide, DRAW4)

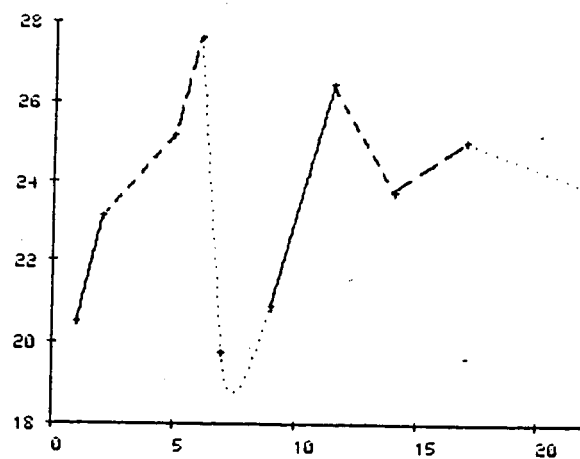
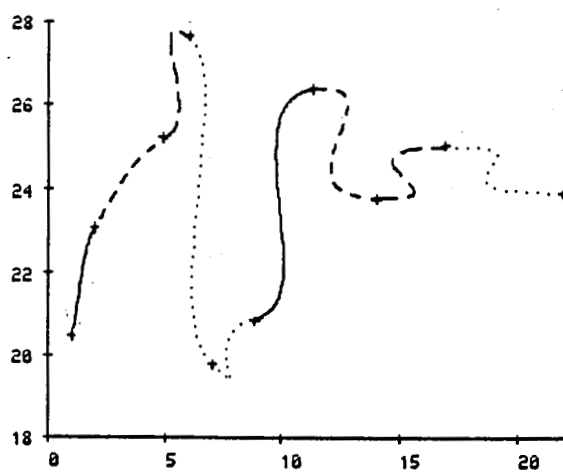


Figure 3. Salinity Conductivity at low tide

DRAW 1 - Method A
 DRAW 2 - Method B
 DRAW 3 - Method B'
 DRAW 4 - Method D

Salinity Conductivity (low tide, Hermite1)



Salinity Conductivity (low tide, Bezier)

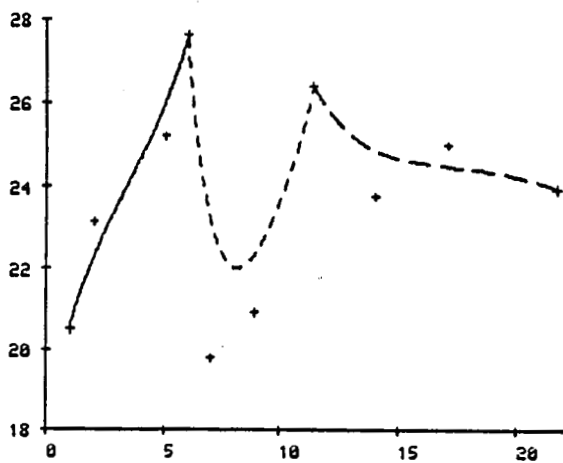
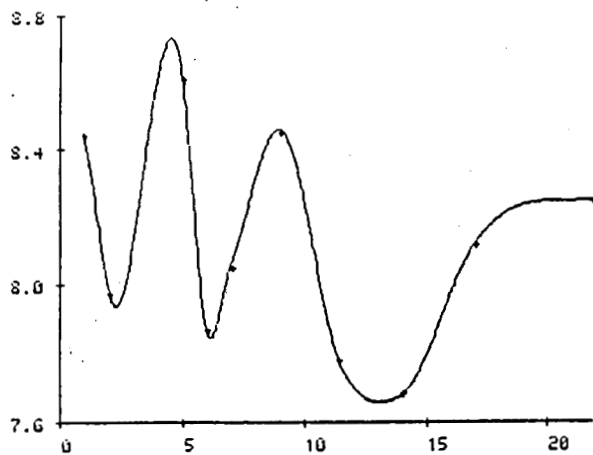
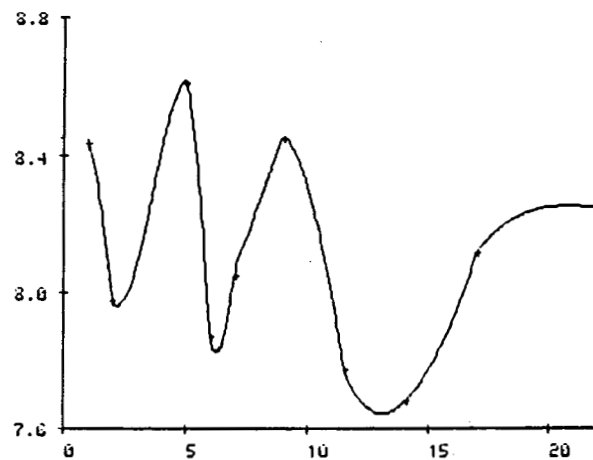


Figure 4. Salinity Conductivity at low tide

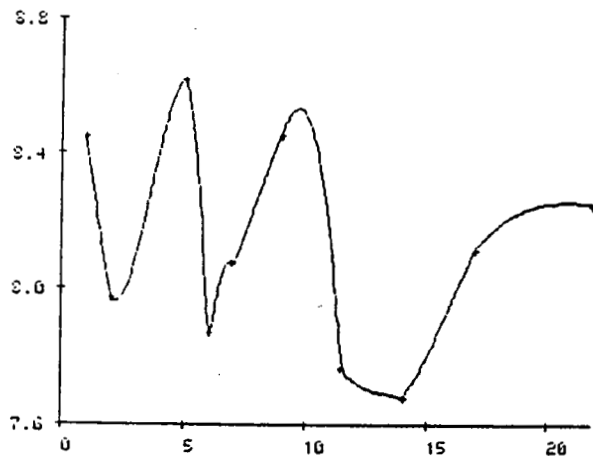
pH (low tide, DRAW1)



pH (low tide, DRAW2)



pH (low tide, DRAW3)



pH (low tide, DRAW4)

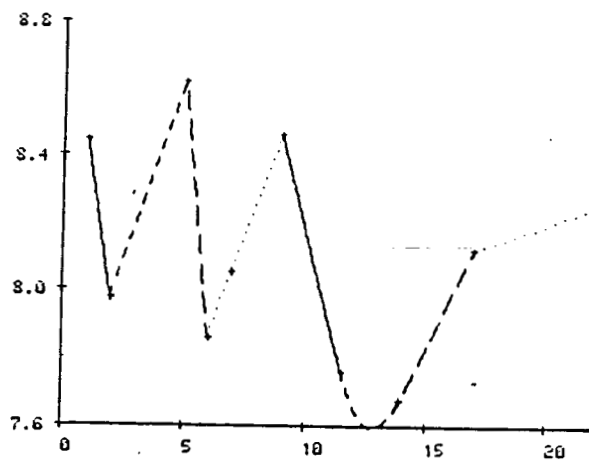
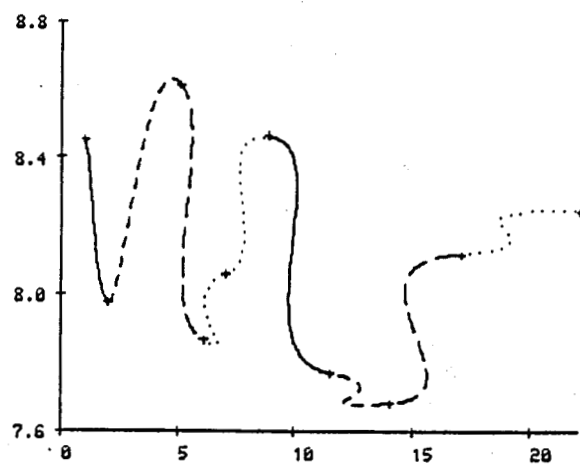


Figure 5. pH at low tide
 DRAW 1 - Method A
 DRAW 2 - Method B
 DRAW 3 - Method B'
 DRAW 4 - Method D

pH (low tide, Hermite1)



pH (low tide, Bezier)

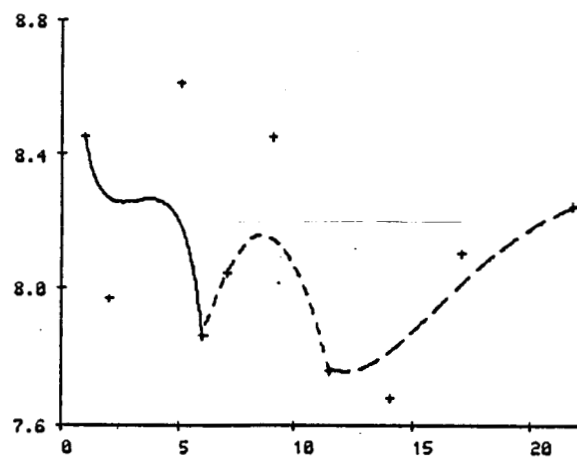
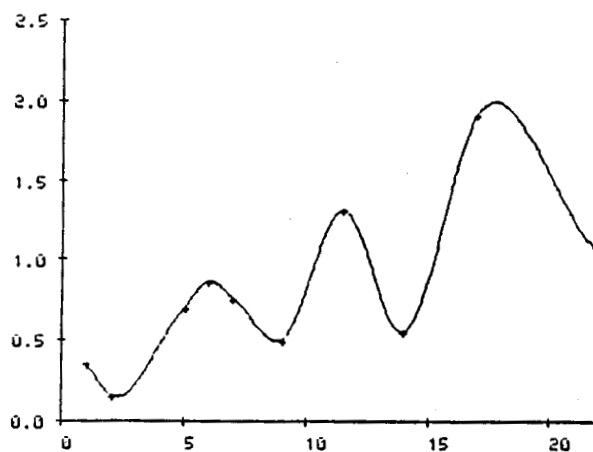
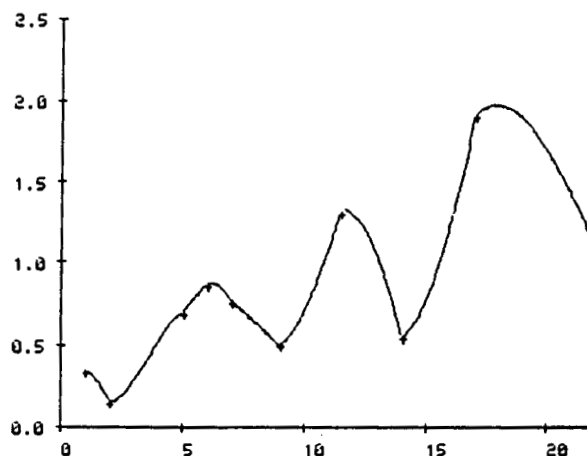


Figure 6. pH at low tide

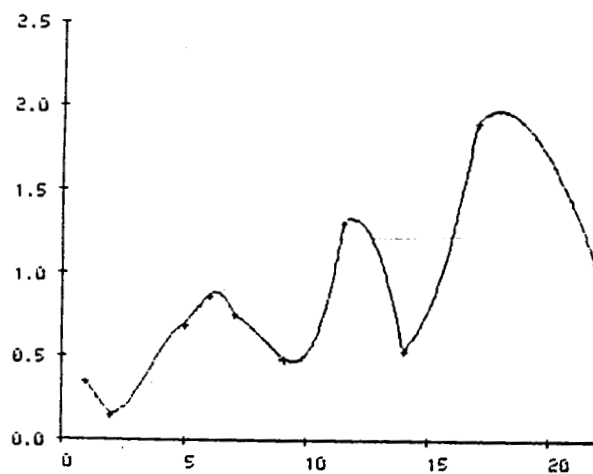
TKN (low tide, DRAW1)



TKN (low tide, DRAW2)



TKN (low tide, DRAW3)



TKN (low tide, DRAW4)

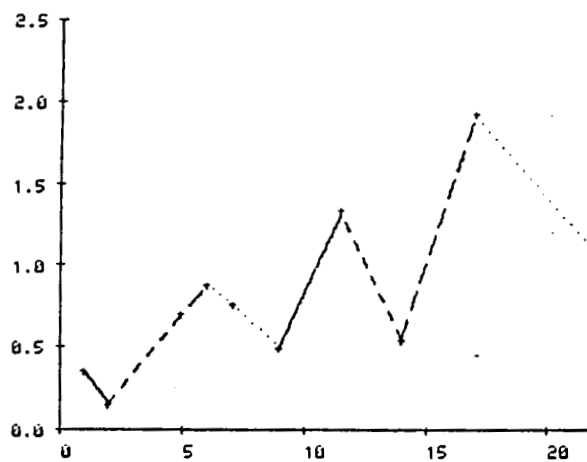
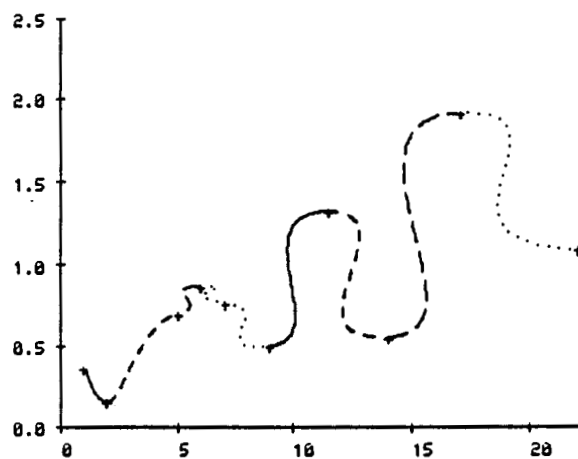


Figure 7. Total Kjeldahl Nitrogen at low tide
 DRAW 1 - Method A
 DRAW 2 - Method B
 DRAW 3 - Method B'
 DRAW 4 - Method D

ORIGINAL PAGE IS
OF POOR QUALITY

TKN (low tide, Hermite)



TKN (low tide, Bezier)

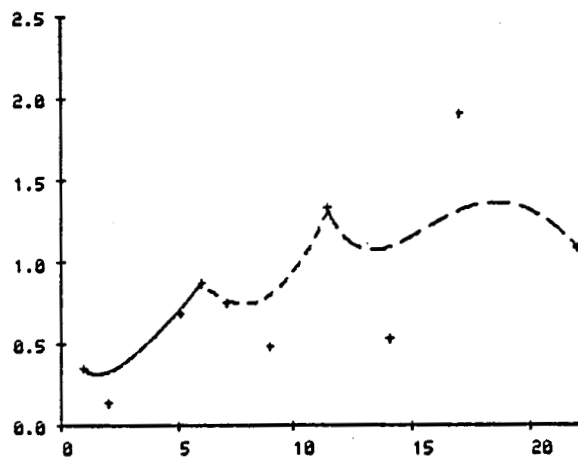
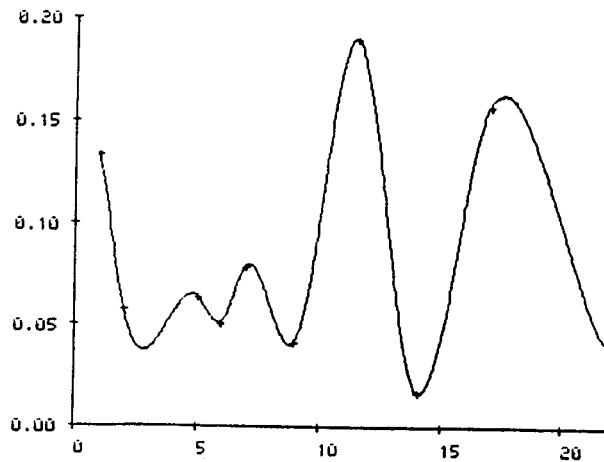


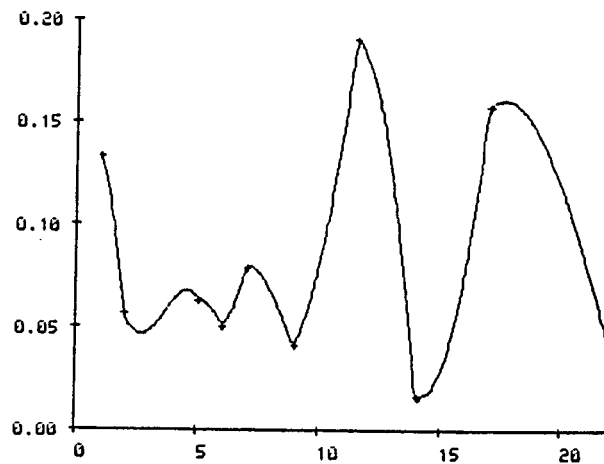
Figure 8. Total Kjeldahl Nitrogen at low tide

ORIGINAL PAGE IS
OF POOR QUALITY

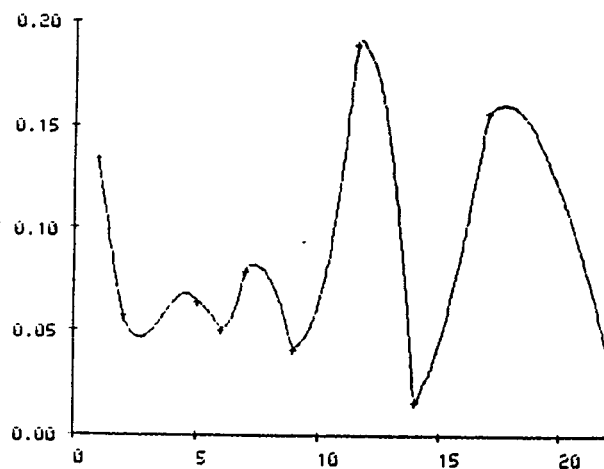
TP04 (low tide, DRAW1)



TP04 (low tide, DRAW2)



TP04 (low tide, DRAW3)



TP04 (low tide, DRAW4)

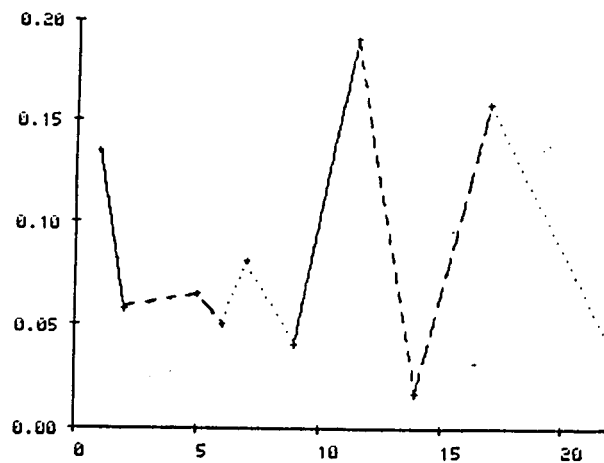
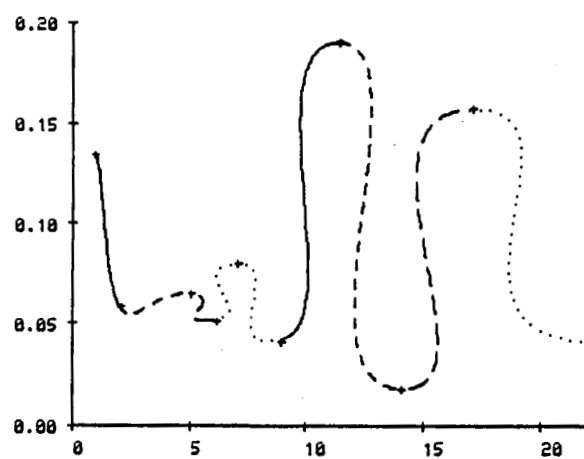


Figure 9. Total Phosphate at low tide
DRAW 1 - Method A
DRAW 2 - Method B
DRAW 3 - Method B'
DRAW 4 - Method D

ORIGINAL PAGE IS
OF POOR QUALITY

TP04 (low tide, Hermitel)



TP04 (low tide, Bezier)

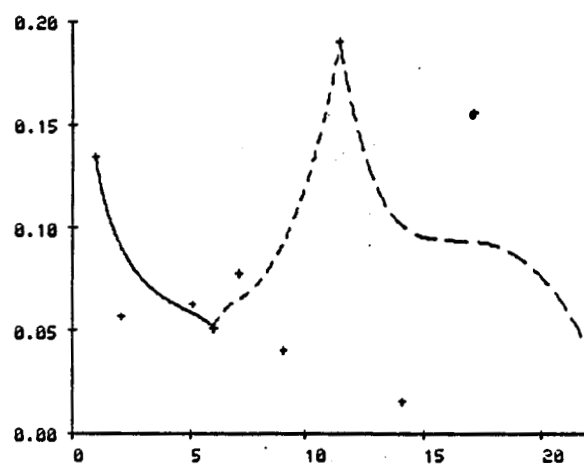
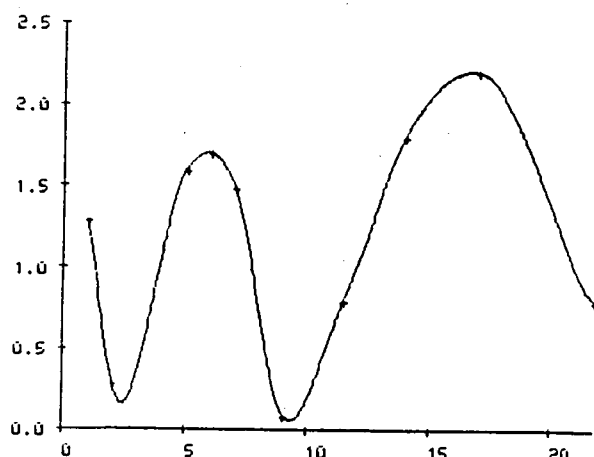


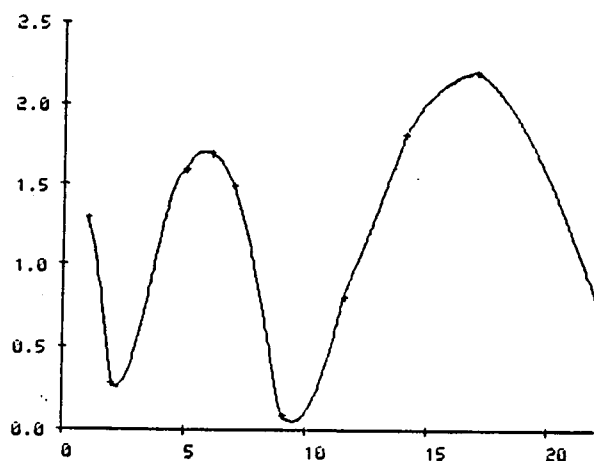
Figure 10. Total Phosphate at low tide

ORIGINAL PAGE IS
OF POOR QUALITY

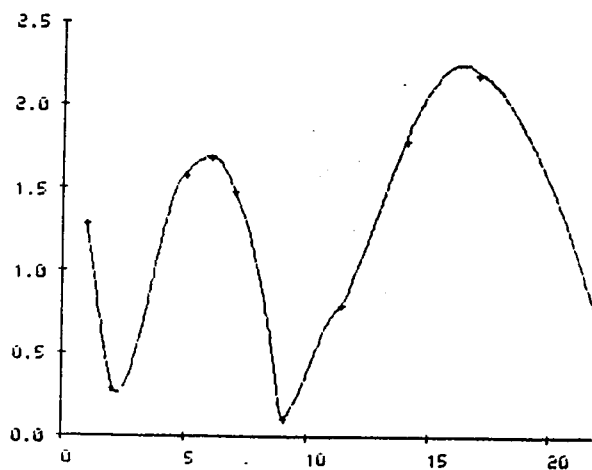
Dissolved Silica (high tide, DRAW1)



Dissolved Silica (high tide, DRAW2)



Dissolved Silica (high tide, DRAW3)



Dissolved Silica (high tide, DRAW4)

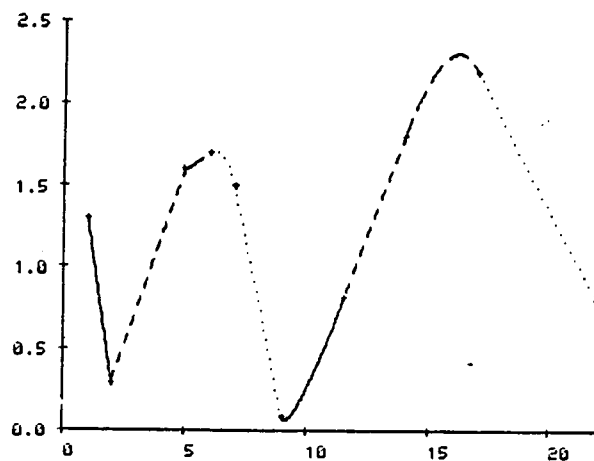
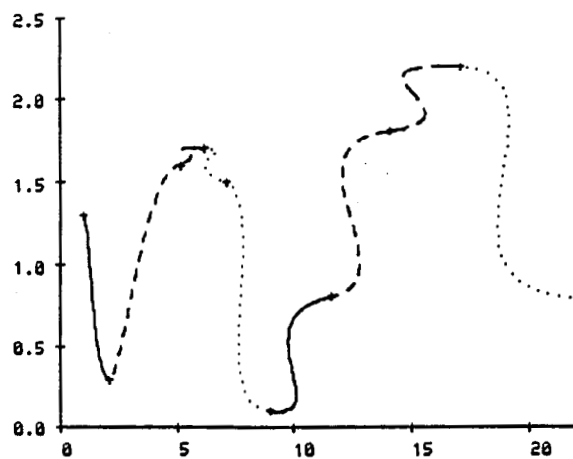


Figure 11. Dissolved Silica at high tide

DRAW 1 - Method A
DRAW 2 - Method B
DRAW 3 - Method B'
DRAW 4 - Method D

ORIGINAL PAGE IS
OF POOR QUALITY

Dissolved Silica (high tide, Hermite1)



Dissolved Silica (high tide, Bezier)

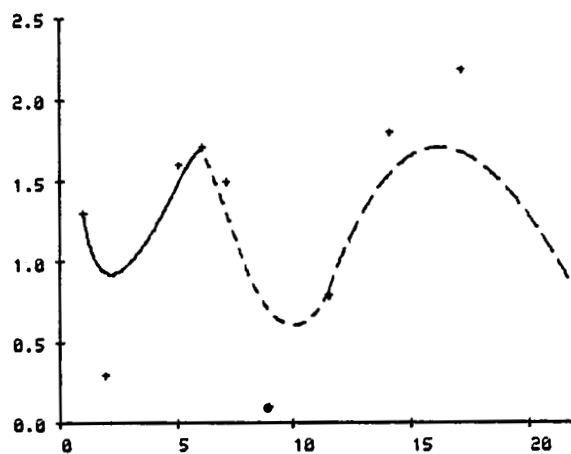
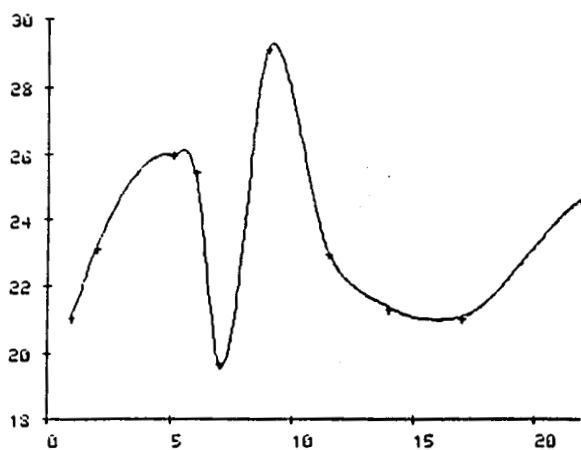
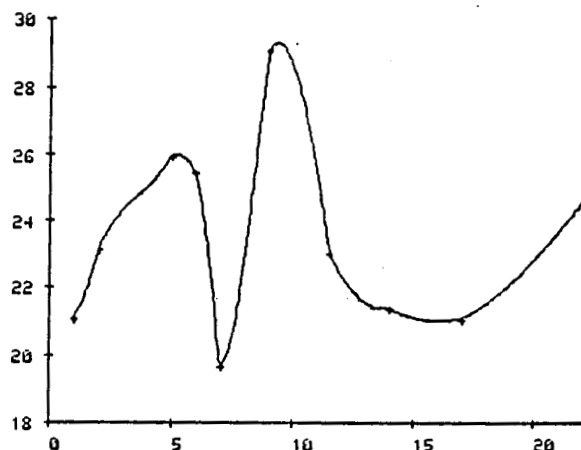


Figure 12. Dissolved Silica at high tide

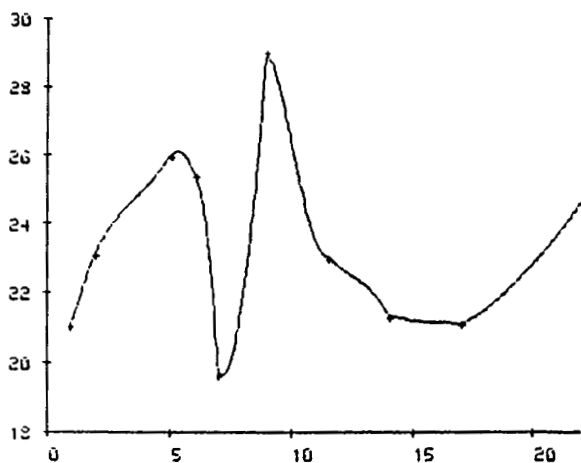
Salinity Conductivity (high tide, DRAW1)



Salinity Conductivity (high tide, DRAW2)



Salinity Conductivity (high tide, DRAW3)



Salinity Conductivity (high tide, DRAW4)

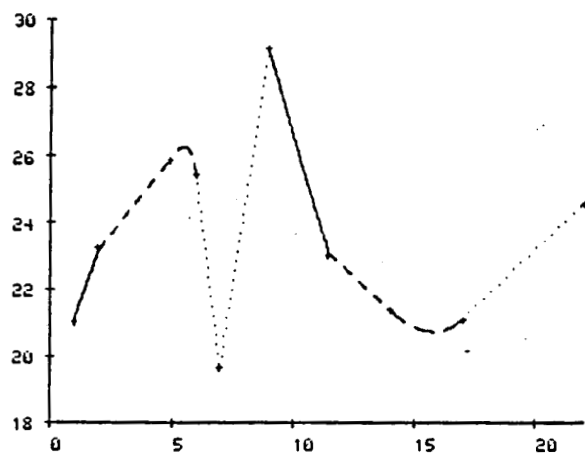
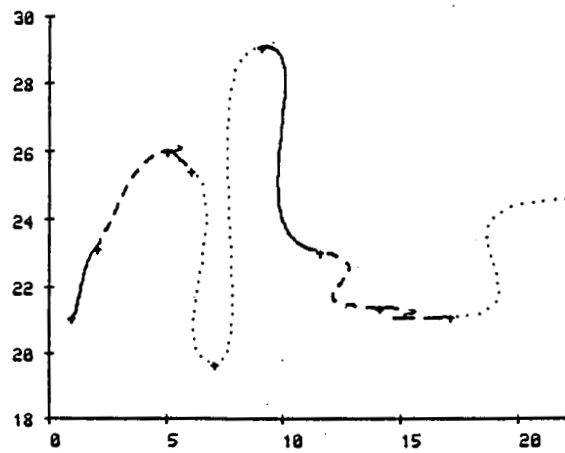


Figure 13. Salinity Conductivity at high tide
 DRAW 1 - Method A
 DRAW 2 - Method B
 DRAW 3 - Method B'
 DRAW 4 - Method D

ORIGINAL PAGE IS
OF POOR QUALITY

Salinity Conductivity (high tide, Hermital)



Salinity Conductivity (high tide, Bezier)

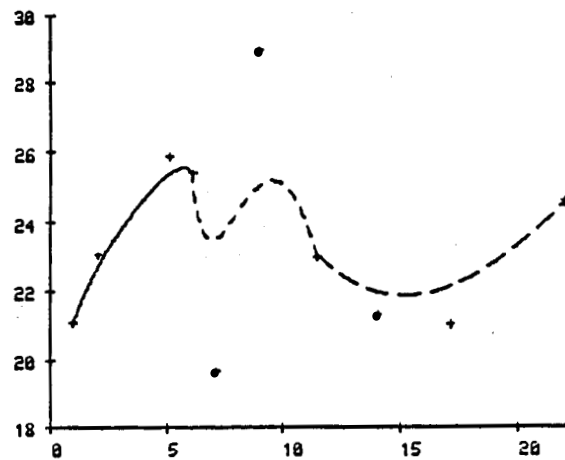
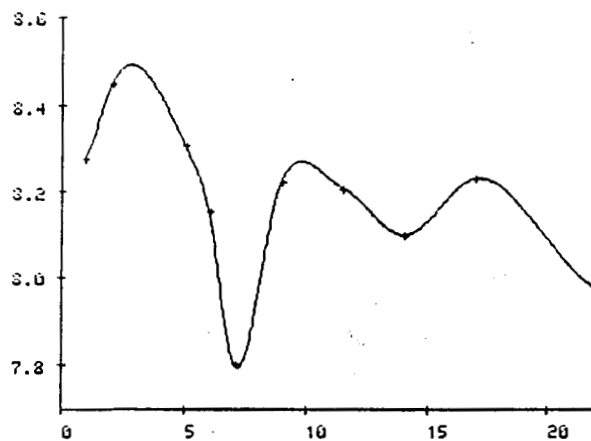
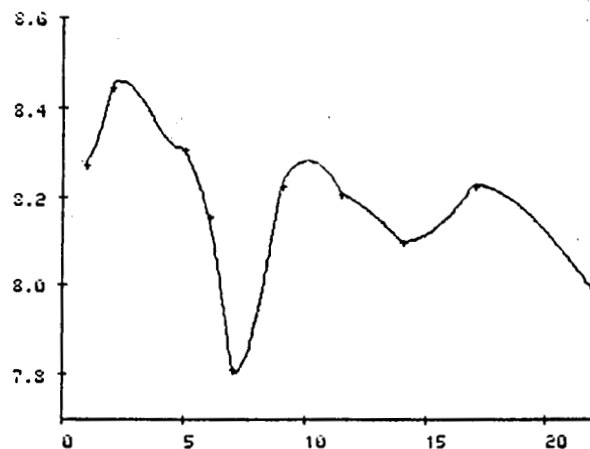


Figure 14. Salinity Conductivity at high tide

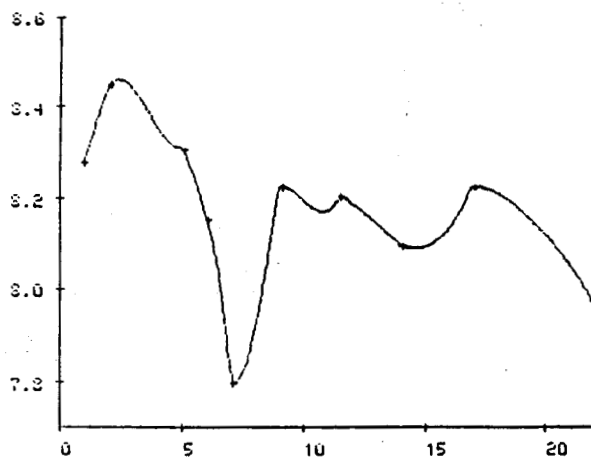
pH (high tide, DRAW1)



pH (high tide, DRAW2)



pH (high tide, DRAW3)



pH (high tide, DRAW4)

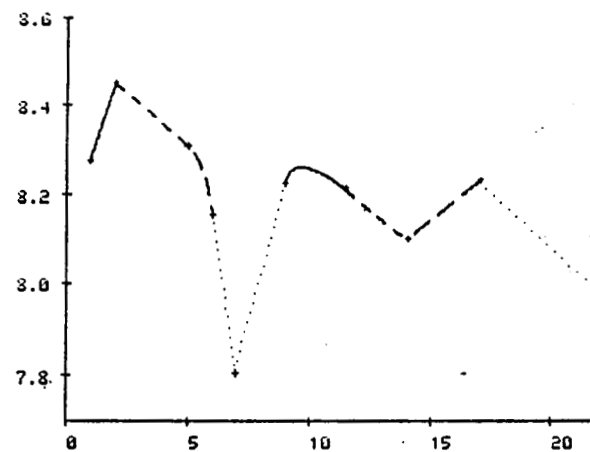
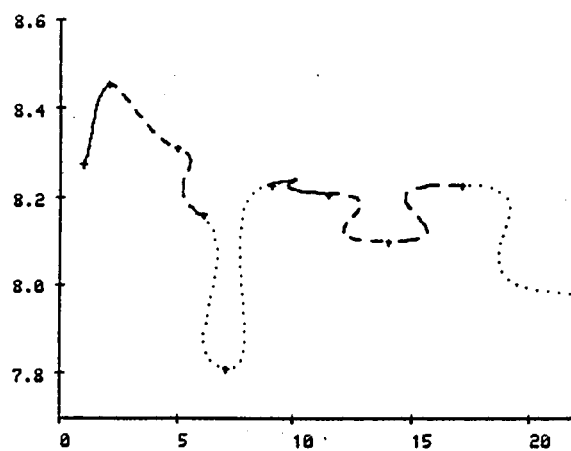


Figure 15. pH at high tide
 DRAW 1 - Method A
 DRAW 2 - Method B
 DRAW 3 - Method B'
 DRAW 4 - Method D

pH (high tide, Hermitel)



pH (high tide, Bezier)

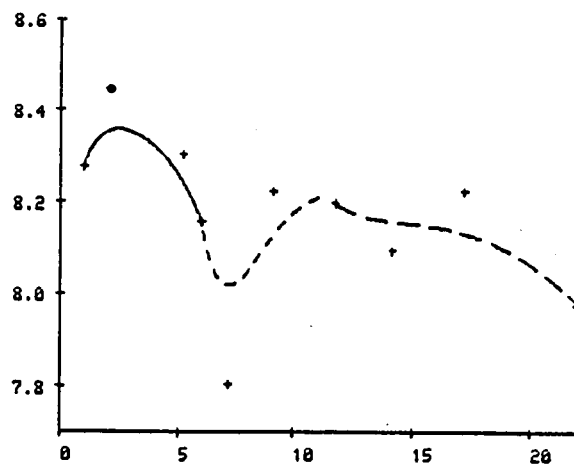
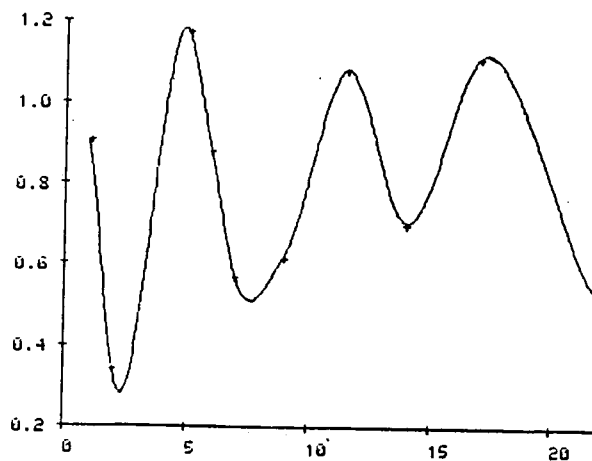


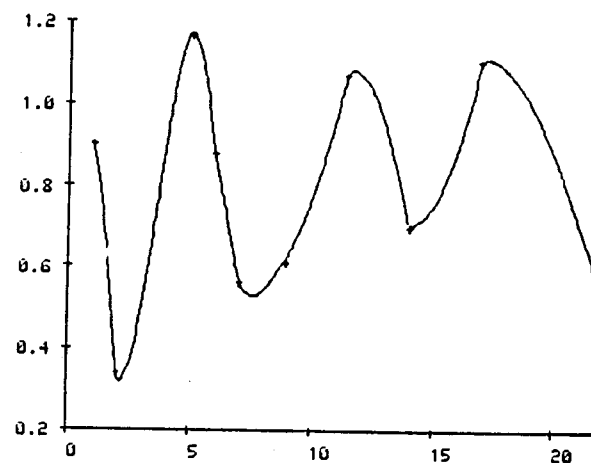
Figure 16. pH at high tide

ORIGINAL PAGE IS
OF POOR QUALITY

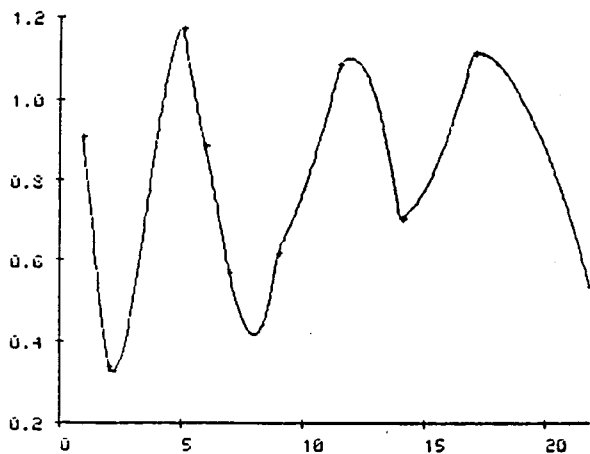
TKN (high tide, DRAW1)



TKN (high tide, DRAW2)



TKN (high tide, DRAW3)



TKN (high tide, DRAW4)

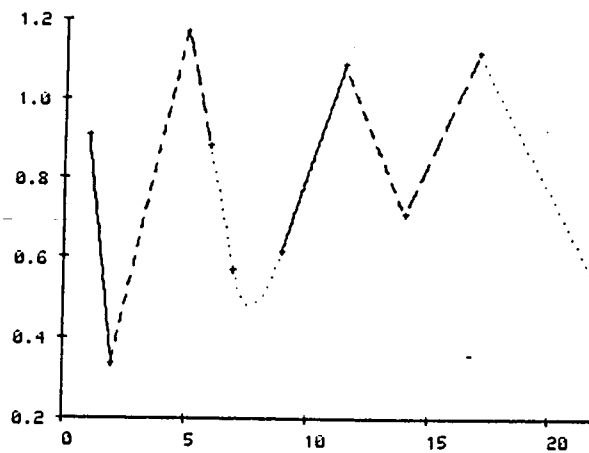
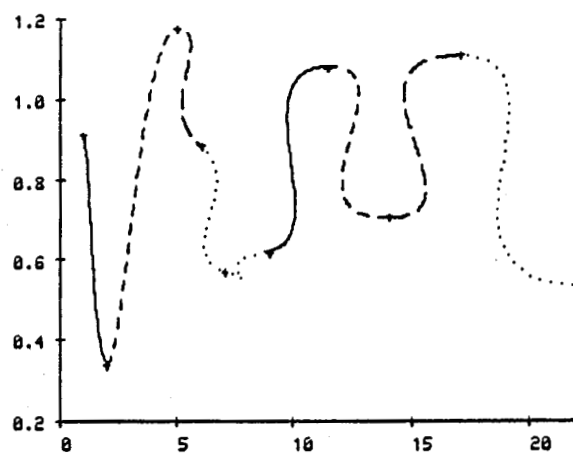


Figure 17. Total Kjeldahl Nitrogen at high tide
DRAW 1 - Method A
DRAW 2 - Method B
DRAW 3 - Method B'
DRAW 4 - Method D

TKN (high tide, Hermite)



TKN (high tide, Bezier)

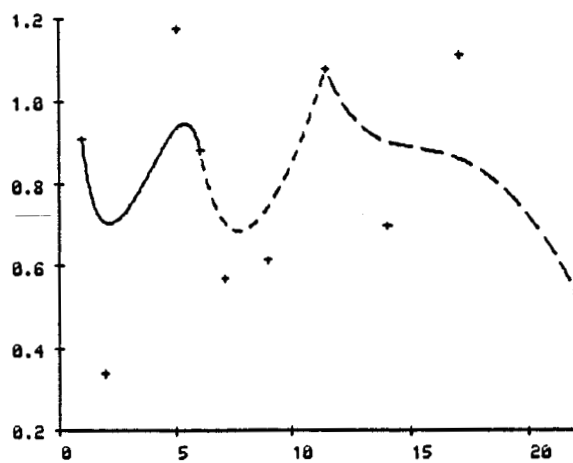
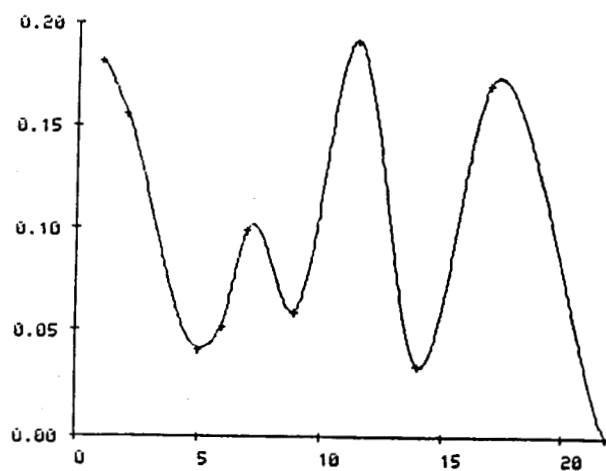
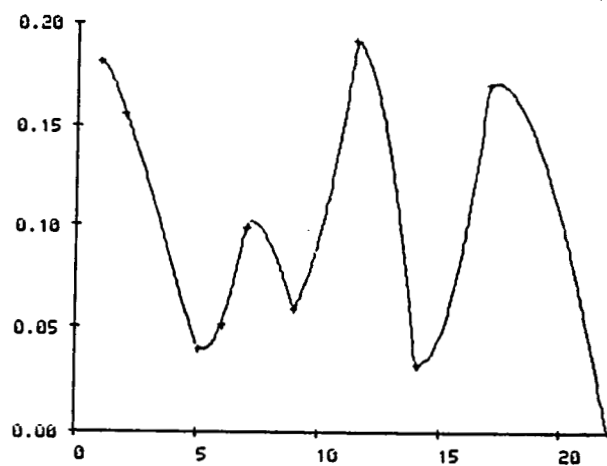


Figure 18. Total Kjeldahl Nitrogen at high tide

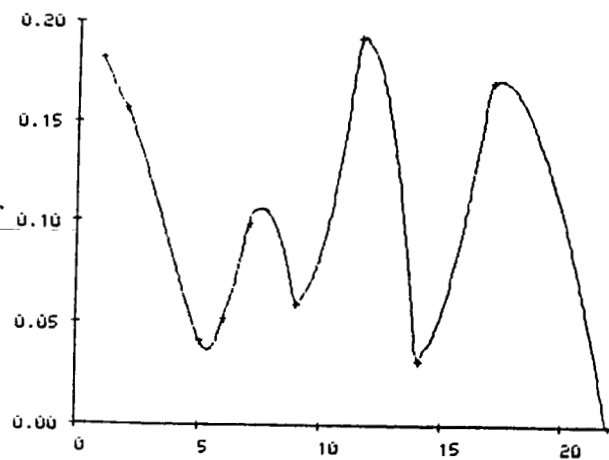
TP04 (high tide, DRAW1)



TP04 (high tide, DRAW2)



TP04 (high tide, DRAW3)



TP04 (high tide, DRAW4)

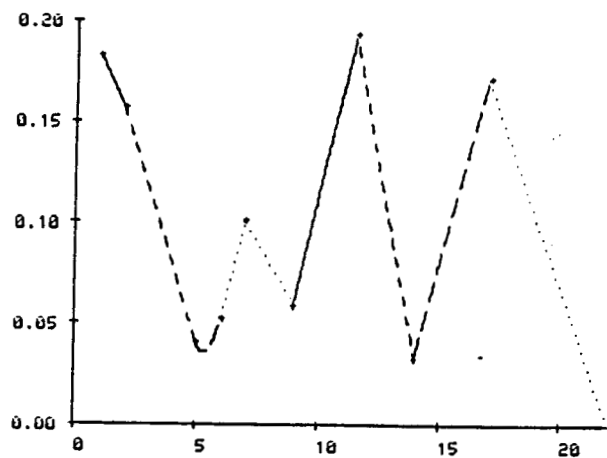
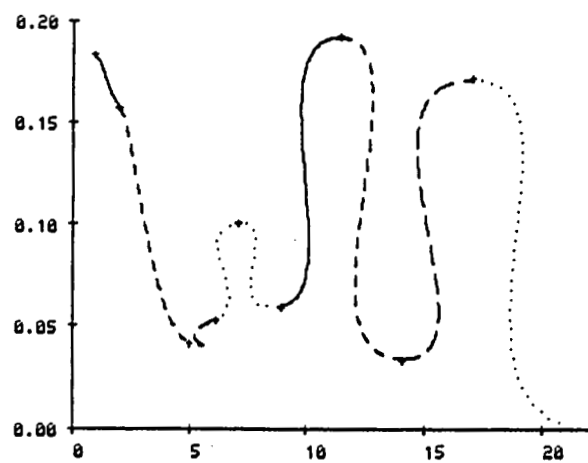


Figure 19. Total Phosphate at high tide
 DRAW 1 - Method A
 DRAW 2 - Method B
 DRAW 3 - Method B'
 DRAW 4 - Method D

TP04 (high tide, Hermite1)



TP04 (high tide, Bezier)

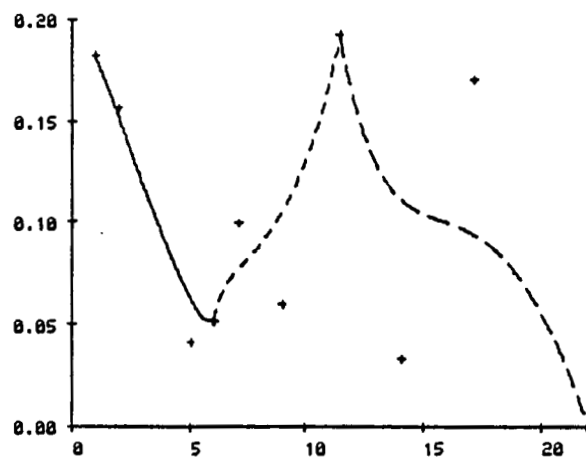


Figure 20. Total Phosphate at high tide

SMP Code:

/* This program uses the osculatory method on page 11 of Brodlie's Mathematical Methods in Computer Graphics and Design to draw a smooth curve that passes through 10 given data points.

An additional point is added to each end of the data. It is the mirror image of the first point (or the last point) of the data.

A quadratic polynomial $q(i,x)=ax^2+bx+c$ defined on $[x(i),x(i+1)]$ is determined. This polynomial passes through points $P(i-1)$, $P(i)$, $P(i+1)$ and is given by

$$\begin{bmatrix} y(i-1) \\ y(i) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} x(i-1)^2 & x(i-1) & 1 \\ x(i)^2 & x(i) & 1 \\ x(i+1)^2 & x(i+1) & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

On $[x(i),x(i+1)]$, blend $q(i,x)$, $q(i+1,x)$ to get $c(i,x)$ by using

$$c(i,x) = \frac{x - x(i+1)}{x(i) - x(i+1)} q(i,x) + \frac{x - x(i)}{x(i+1) - x(i)} q(i+1,x)$$

Join these $\{c(i,x) : i = 2,3,\dots,10\}$ into a piecewise smooth curve */

x : {-1,1,2,5,6,7,9,11.5,14,17,22,27}

```
Do[i,2,11, \
  t1 : {{y[i-1]},{y[i]},{y[i+1]}}; \
  t2 : {{x[i-1]^2,x[i-1],1},{x[i]^2,x[i],1},{x[i+1]^2,x[i+1],1}}; \
  coef : Mdiv[t1,t2]; \
  q[i,$u] : coef[1,1] ($u)^2 + coef[2,1] $u + coef[3,1]]
```

```
Do[j,2,10, \
  c[j,$u] : ((($u - x[j+1])/(x[j] - x[j+1])) q[j,$u] + \
  ((($u - x[j])/(x[j+1] - x[j])) q[j+1,$u])
```

```
f[$u] : c[2,$u]
f[$u_=$u>2] : c[3,$u]
f[$u_=$u>5] : c[4,$u]
f[$u_=$u>6] : c[5,$u]
f[$u_=$u>7] : c[6,$u]
f[$u_=$u>9] : c[7,$u]
f[$u_=$u>11.5] : c[8,$u]
f[$u_=$u>14] : c[9,$u]
f[$u_=$u>17] : c[10,$u]
Graph[Smp][1]:0
Graph1[Smp][1]:0
Graph[f[t],t,1,22,,,{0,22}]
```

```

/* Method B */
x : {-1,1,2,5,6,7,9,11.5,14,17,22,27}

/* Estimate slope at x(i) */
Do[i,2,12, \
  m[i] : (y[i] - y[i-1])/(x[i] - x[i-1])]
Do[j,2,11, \
  s[j] : 0.5 (m[j] + m[j+1])]

/* Determine cubic polynomials c(i,x) */
Do[k,2,10, \
  t1 : {{y[k-1]}, {y[k]}, {y[k+1]}, {s[k]}} ; \
  t2 : {{x[k-1]^3, x[k-1]^2, x[k-1], 1}, \
        {x[k]^3, x[k]^2, x[k], 1}, \
        {x[k+1]^3, x[k+1]^2, x[k+1], 1}, \
        {3*x[k]^2, 2*x[k], 1, 0}} ; \
  coef : Mdiv[t1,t2] ; \
  c[k,$u] : coef[1,1] ($u)^3 + coef[2,1] ($u)^2 + \
            coef[3,1] $u + coef[4,1]]

/* piece c(i,x) together */
f[$u] : c[2,$u]
f[$u_=$u>2] : c[3,$u]
f[$u_=$u>5] : c[4,$u]
f[$u_=$u>6] : c[5,$u]
f[$u_=$u>7] : c[6,$u]
f[$u_=$u>9] : c[7,$u]
f[$u_=$u>11.5] : c[8,$u]
f[$u_=$u>14] : c[9,$u]
f[$u_=$u>17] : c[10,$u]
Graph[Smp][1] : 0
Graph[f[t],t,1,22,,,{0,22}]

```



```

/* Method B' */

x : {-1,1,2,5,6,7,9,11.5,14,17,22,27}

/* Calculate Akima's slope estimate at x(i) */
Do[i,2,12, \
  m[i] : (y[i] - y[i-1])/(x[i] - x[i-1])
  s[2] : m[3]
  s[3] : 0.5 (m[3] + m[4])
Do[j,4,9, \
  s[j] : (Abs[m[j+2] - m[j+1]] m[j] + Abs[m[j-1] - m[j]] * \
    m[j+1]) / (Abs[m[j+2] - m[j+1]] + Abs[m[j-1] - m[j]])
  s[10] : 0.5 (m[10] + m[11])
  s[11] : m[11]

/* Define cubic polynomials c(i,x) */
Do[k,2,10, \
  t1 : {{y[k-1]}, {y[k]}, {y[k+1]}, {s[k]}} ; \
  t2 : {{x[k-1]^3, x[k-1]^2, x[k-1], 1}, \
    {x[k]^3, x[k]^2, x[k], 1}, \
    {x[k+1]^3, x[k+1]^2, x[k+1], 1}, \
    {3*x[k]^2, 2*x[k], 1, 0}} ; \
  coef : Mdiv[t1,t2] ; \
  c[k,$u] : coef[1,1] ($u)^3 + coef[2,1] ($u)^2 + \
    coef[3,1] $u + coef[4,1]]

/* Piece c(i,x) together */
f[$u] : c[2,$u]
f[$u_=$u>2] : c[3,$u]
f[$u_=$u>5] : c[4,$u]
f[$u_=$u>6] : c[5,$u]
f[$u_=$u>7] : c[6,$u]
f[$u_=$u>9] : c[7,$u]
f[$u_=$u>11.5] : c[8,$u]
f[$u_=$u>14] : c[9,$u]
f[$u_=$u>17] : c[10,$u]
Graph[Smp][1] : 0
Graph[f[t],t,1,22]

```

```

/* Cubic hermite form */
x : {-1,1,2,5,6,7,9,11.5,14,17,22,27}

/* Estimate first order derivative at x(i) */
Do[i,2,12, \
  m[i] : (y[i] - y[i-1])/(x[i] - x[i-1])]
Do[j,2,11, \
  s[j] : 0.5 (m[j] + m[j+1])]

p : {}
Do[k,1,11, p : Cat[p, {{x[k],y[k]} }]]

q : {}
Do[k,1,11, q : Cat[q, {{x[k],s[k]} }]]

/* define the cubic polynomials */
Do[k,2,10, \
  f[k,$u] : (2 ($u)^3 - 3 ($u)^2 + 1) p[k] + \
            (-2 ($u)^3 + 3 ($u)^2) p[k+1] + \
            (($u)^3 - 2 ($u)^2 + $u) q[k] + \
            (($u)^3 - ($u)^2) q[k+1]]

Graph[{f[2,t],f[3,t],f[4,t],f[5,t],f[6,t],f[7,t], \
  f[8,t],f[9,t],f[10,t]},t,0,1]

```

```

/* McLaughlin's Method */

delta[$i] : p[$i] - p[$i-1]

/* Determine the regularity of the first point */
regular1[$p] : [ a : $p[1] - $p[2] ; \
                 b : $p[3] - $p[2] ; \
                 a.b > 0 ]

/* Determine the regularity of the last point */
regularN[$p,$n] : [ a : $p[$n-2] - $p[$n-1] ; \
                    b : $p[$n] - $p[$n-1] ; \
                    a.b > 0 ]

/* Determine the regularity of the in between points */
regular[$p,$i] : \
[ d : Det[{{Elem[delta[$i],{1}], Elem[delta[$i+2],{1}]}}, \
           {Elem[delta[$i],{2}], Elem[delta[$i+2],{2}]}}] ; \
  d1 : Det[{{Elem[delta[$i],{1}], Elem[delta[$i+1],{1}]}, \
            {Elem[delta[$i],{2}], Elem[delta[$i+1],{2}]}}] + \
        {Elem[delta[$i],{2}], Elem[delta[$i+1],{2}]}, \
        {Elem[delta[$i],{1}], Elem[delta[$i+1],{1}]}}] ; \
  d2 : Det[{{Elem[delta[$i+2],{1}], Elem[delta[$i+1],{1}]}, \
            {Elem[delta[$i+2],{2}], Elem[delta[$i+1],{2}]}}] + \
        {Elem[delta[$i+2],{2}], Elem[delta[$i+1],{2}]}, \
        {Elem[delta[$i+2],{1}], Elem[delta[$i+1],{1}]}}] ; \
  del[$i] : d ; \
  dels[$i] : d1 ; \
  And[d != 0, Sign[d] = Sign[d1], Sign[d] = Sign[d2]]]

norm[$a,$b] : ($a[1] - $b[1])^2 + ($a[2] - $b[2])^2

/* the set of data points P has been batched in already */

q : 0.5
n : 10

r : {}
it : {}
t : {}

r1 : List[regular1[p]]
r : Cat[r,r1]
r1 : List[regular[p,2]]
r : Cat[r,r1]

t[1] : If[ r[1]=1, \
           [If[ r[2]=1, \
               [a : p[2] + s*(p[3] - p[2]) ; \
                e1 : (a - p[2]).(p[1] - p[2]) = \
                    (a - p[1]).(p[2] - p[1]) ; \
                e2 : norm[a,p[2]] = norm[a,p[1]] ; \
                Sol[{e1,e2},s] ; \
               ]
           ]

```

```

it[1] : p[2] + s*(p[3] - p[2]) ; \
a : p[1] + q*(it[1] - p[1]) ; \
b : p[2] + h*(a - p[2]) ; \
f1 : (b - p[2])*(p[1] - p[2]) = \
      (b - p[1])*(p[2] - p[1]) ; \
f2 : norm[b,p[2]] = norm[b,p[1]] ; \
Sol[{f1,f2},h] ; \
p[2] + h*(a - p[2]) , \
[ a : p[2] + s*(p[3] - p[2]) ; \
  e1 : (a - p[2])*(p[1] - p[2]) = \
        (a - p[1])*(p[2] - p[1]) ; \
  e2 : norm[a,p[2]] = norm[a,p[1]] ; \
  Sol[{e1,e2},s] ; \
  it[1] : p[2] + s*(p[3] - p[2]) ; \
  it[1]]], \
p[1]]

Do[ j,2,n-2, \
  r1 : If[ j<n-2, regular[p,j+1] , \
           regularN[p,n]] ; \
  r2 : List[r1] ;
  r : Cat[r,r2];
  If[ r[j]=1, \
    [ If[ r[j+1]=1, \
      [ s : 1 + dels[j] / del[j] ; \
        it[j] : p[j+2] + s*(p[j+1] - p[j+2]) ; \
        a : p[j] + q*(it[j] - p[j]) ; \
        b : p[j+1] + s*(a - p[j+1]) ; \
        c : p[j] + h*(t[j-1] - p[j]) ; \
        Sol[b=c,h] ; \
        t[j] : p[j] + h*(t[j-1] - p[j])], \
      [ s : 1 + dels[j] / del[j] ; \
        t[j] : p[j+2] + s*(p[j+1] - p[j+2])]]], \
    t[j] : p[j]]]

t[n-1] : If[ r[n-1]=1, \
  [ a : p[n-1] + s*(t[n-2] - p[n-1]) ; \
    e1 : (a - p[n-1])*(p[n] - p[n-1]) = \
          (a - p[n])*(p[n-1] - p[n]) ; \
    e2 : norm[a,p[n-1]] = norm[a,p[n]] ; \
    Sol[{e1,e2},s] ; \
    p[n-1] + s*(t[n-2] - p[n-1])], \
  p[n-1]]

Do[ j,1,n-1, \
  f[j,$u] : \
  If[r[j]=1, $u^2 p[j+1] + 2 $u (1-$u) t[j] + \
             (1-$u)^2 p[j], \
    p[j] + $u (p[j+1] - p[j])]

Graph[{f[1,$u],f[2,$u],f[3,$u],f[4,$u],f[5,$u],f[6,$u], \
       f[7,$u],f[8,$u],f[9,$u]}, $u,0,1]

```

```

/* Bezier curve using 4 control points */
x : {1,2,5,6,7,9,11.5,14,17,22}
p : {}
Do[i,1,10, p : Cat[p, {{x[i],y[i]} }]]

Do[i,1,7,3, \
  f[i,$u] : Sum[Comb[3,k] ($u)^k (1 - $u)^(3-k) p[i+k], {k,0,3}]]

q : Map[Pt,p]
Graph[{f[1,t],f[4,t],f[7,t]},t,0,1]

```

IV. KINEMATICS OF ROBOT ARM

Introduction:

The purpose of this research work is to study the kinematics of a single link robot arm. In the solution of equations of motion of this system, the symbolic manipulation programs MACSYMA and SMP have been used. Also, some IMSL subroutines are also utilized.

The Lagrangian of the system is set-up in a fixed frame of reference. Hamiltonian is calculated. From this, Hamilton-Jacobi equation is obtained. Solution of this equation gives the time evolution of the system.

Definition of symbols:

Before setting-up the necessary equations, the following list of symbols is defined. Note that subscript 1 denotes the first rod and subscript 2 denotes the second rod. Subscripts to I denote the appropriate axes about which the moment of inertia of a particular rod is defined. See Figures 1 and 2.

L = Lagrangian

T = Kinetic Energy

V = potential energy

I = moment of inertia

ϕ = angle of rotation

m = mass

g = gravitational acceleration

h = height

w = angular speed

l = distance of CM from axis of rotation

Lagrange's Equations:

The Lagrangian of the first rod is

$$L_1 = T_1 - V_1$$

$$= \frac{1}{2} I_{1Z_1Z_1} \dot{\phi}_1^2 - m_1 g h_1 .$$

Kinetic energy T_2 of second rod in the $X_2Y_2Z_2$ system is

$$T_2 = \frac{1}{2} I_{2Y_2Y_2} \dot{\phi}_2^2 .$$

Now the vector is

$$\vec{\phi}_2 = \phi_2 \hat{Y}_2 .$$

In $X_1Y_1Z_1$ system, the vector $\dot{\vec{\phi}}_2$ is given by

$$(d\vec{\phi}_2/dt)_{X_1Y_1Z_1} = (d\vec{\phi}_2/dt)_{X_2Y_2Z_2} + \vec{\omega}_1 \times \vec{\phi}_2$$

This can be shown to give

$$(\dot{\phi}_2^2)_{X_1Y_1Z_1} = \dot{\phi}_2^2 + \dot{\phi}_1^2 \phi_2^2 .$$

Therefore,

T_2 in $X_1Y_1Z_1$ system is

$$(T_2)_{X_1Y_1Z_1} = \frac{1}{2} I_{2Y_2Y_2} \dot{\phi}_2^2 + \frac{1}{2} I_{2Y_2Y_2} \dot{\phi}_1^2 \phi_2^2$$

and

$$(V_2)_{X_1 Y_1 Z_1} = m_2 g h_2 .$$

Kinetic energy of motion of CM of rod 2 about Z_1 axis is

$$(T_2)_{Z_1} = \frac{1}{2} m_2 l_2^2 \dot{\phi}_1^2 .$$

From above, the Lagrangian of second rod in the $X_1 Y_1 Z_1$ system is

$$\begin{aligned} L_2 = & \frac{1}{2} m_2 l_2^2 \dot{\phi}_1^2 + \frac{1}{2} I_{2Y_2 Y_2} \dot{\phi}_2^2 \\ & + \frac{1}{2} I_{2Y_2 Y_2} \dot{\phi}_1^2 \dot{\phi}_2^2 - m_2 g h_2 . \end{aligned}$$

Total Lagrangian can be shown to be

$$\begin{aligned} L = L_1 + L_2 \\ = \frac{1}{2} I_1 \dot{\phi}_1^2 + \frac{1}{2} I_2 \dot{\phi}_2^2 + \frac{1}{2} I_2 \dot{\phi}_1^2 \dot{\phi}_2^2 - V \end{aligned}$$

where

$$I_1 = \frac{1}{2} (I_{1Z_1 Z_1} + m_2 l_2^2)$$

$$I_2 = \frac{1}{2} I_{2Y_2 Y_2}$$

$$V = g(m_1 h_1 + m_2 h_2) .$$

Lagrange's equations of motion are

$$\ddot{\phi}_1 = - \frac{2 \phi_2 \dot{\phi}_1 \dot{\phi}_2}{(a + \phi_2^2)}$$

$$\ddot{\phi}_2 = \phi_2 \dot{\phi}_1^2$$

with

$$a = I_1/I_2.$$

Hamiltonian Formulation:

Let the generalized coordinates be $\vec{q} = (\phi_1, \phi_2)$. Generalized momenta are vector \vec{p} .

Now,

$$p_1 = \partial L / \partial \dot{q}_1 = I_1 \dot{\phi}_1 + I_2 \dot{\phi}_1 \phi_2^2$$

$$p_2 = \partial L / \partial \dot{q}_2 = I_2 \dot{\phi}_2.$$

Hamiltonian H is

$$H = \sum_{i=1}^2 q_i p_i - L$$

$$= \frac{1}{2} I_1 \dot{\phi}_1^2 + \frac{1}{2} I_2 \dot{\phi}_2^2 + \frac{1}{2} I_2 \dot{\phi}_1^2 \phi_2^2 + V$$

$$= p_1^2 / (2(I_1 + I_2 \phi_2^2)) + p_2^2 / (2I_2) + V.$$

Hamilton's equations of motion are

$$\dot{\phi}_1 = \partial H / \partial p_1 = p_1 / (I_1 + I_2 \phi_2^2)$$

$$\dot{\phi}_2 = \partial H / \partial p_2 = p_2 / I_2$$

$$\dot{p}_1 = - \partial H / \partial q_1 = 0$$

$$\dot{p}_2 = - \partial H / \partial q_2 = I_2 \phi_2 p_1^2 / (I_1 + I_2 \phi_2^2)^2.$$

Note that $p_1 = \text{constant}$.

If one assumes the system to be non-dissipative, $H = E = \text{constant}$. In this case,

$$p_2 = \pm A^{\frac{1}{2}} \left(1 - \frac{C}{1 + \phi_2^2} \right)^{\frac{1}{2}},$$

remembering that $p_1 = \text{constant}$ and defining

$$C = B/A \geq 0.$$

with

$$A = 2(E-V)$$

and

$$B = \gamma^2 (= p_1^2) > 0.$$

This leads to the interesting phenomena of bifurcation. When $0 \leq C \leq 1$, then $-\infty \leq \phi_2 \leq \infty$. This corresponds to rotation. When $C > 1$, $\phi_2 \leq -(C-1)^{\frac{1}{2}}$. This corresponds to libration. Thus $C = 1$ is a bifurcation point.

Hamilton-Jacobi Equation:

Redefine the momenta and coordinates as

$$\tilde{p}_1 = p_1 / I_1^{\frac{1}{2}}$$

$$\tilde{p}_2 = p_2 / I_2^{\frac{1}{2}}$$

$$\tilde{\phi}_2 = \phi_2 / (I_1 / I_2)^{\frac{1}{2}} .$$

From now on, p_1 , p_2 and ϕ_2 will denote \tilde{p}_1 , \tilde{p}_2 and $\tilde{\phi}_2$. With this redefinition,

$$H = \frac{p_1^2}{2(1 + \phi_2^2)} + \frac{p_2^2}{2} + V.$$

Again, assuming the system to be conservative, Hamilton-Jacobi equation will be given by

$$\frac{\gamma^2}{2(1 + \phi_2^2)} + \frac{p_2^2}{2} + V = E$$

with

$$p_2 = \frac{\partial W}{\partial \phi_2} .$$

From Hamilton-Jacobi theory, one finds that

$$W = \gamma \phi_1 + \int \left(A - \frac{B}{(1 + \phi_2^2)} \right)^{\frac{1}{2}} d\phi_2 .$$

This leads to

$$\begin{aligned} Q_1 &= t + \beta_1 \\ &= \partial W / \partial E \\ &= \int d\phi_2 \frac{1}{\left[A - \frac{B}{1+\phi_2^2}\right]^{\frac{1}{2}}} . \end{aligned}$$

β_1 is to be determined from initial condition on ϕ_2 .

Similarly,

$$\begin{aligned} Q_2 &= \beta_2 \\ &= \partial W / \partial \gamma \end{aligned}$$

which yields

$$\phi_1 = \beta_2 + \gamma \int \frac{d\phi_2}{(1 + \phi_2^2) \left[A - \frac{B}{1 + \phi_2^2}\right]^{\frac{1}{2}}} .$$

Again, β_2 has to be determined from initial condition on ϕ_1 .

In terms of $C = B/A$, one obtains

$$t + \beta_1 = \frac{1}{A^{\frac{1}{2}}} \int \frac{d\phi_2}{\left[1 - \frac{C}{(1 + \phi_2^2)}\right]^{\frac{1}{2}}}$$

and

$$\phi_1 = \beta_2 + \frac{\gamma}{A^{\frac{1}{2}}} \int \frac{d\phi_2}{(1 + \phi_2^2) \left[1 - \frac{C}{1 + \phi_2^2}\right]^{\frac{1}{2}}} .$$

Evaluation of integrals:

If one makes a change of variable from ϕ_2 to θ with $\phi_2 = \cot \theta$, the first integral reduces to

$$-F(\theta, K) + E(\theta, K) + \Delta \cot \theta$$

where

$$k^2 = C$$

$$\Delta = [1 - C \sin^2 \theta]^{\frac{1}{2}}$$

and

F = Elliptical integral of first kind

E = Elliptical integral of second kind.

The second integral can similarly be evaluated to $-F(\theta, k)$.

This completes the solution of the equation of motion of system. With the evaluation of these integrals, one can obtain $t(\phi_2)$. This can be inverted to give $\phi_2(t)$. Once $\phi_2(t)$ is known, $\phi_1(t)$ can be obtained from second integral.

Use of MACSYMA:

The MACSYMA program which obtains the Hamilton-Jacobi equation is shown on the following pages. In Figure 3, ϕ_2 is shown as a function of time t . To evaluate the integrals, first SMP was used. But in doing so, some serious errors in the elliptical functions evaluation programs of SMP were discovered. Inference Corporation was informed this. SMP has now corrected these errors. Since SMP could not be used to evaluate the elliptical functions, the subroutines MMLINF, MMLIND and MMLINJ of IMSL were utilized.

ORIGINAL PAGE IS
OF POOR QUALITY

Nov 17 11:42 1981 Hamilton Page 1

(d1)

Hamilton

(c2) depends([l1,[a1,a2],[h],[a1,a2;p1;p2],[a1,a2;p1;p2],[t]]

(c3) l1(a1,a2), l(a1,a2,p1,p2), a1(t), a2(t), p1(t), p2(t)

(c4) diff(l,a1);

(d3)

da1

dt

(c4) diff(a1,t);

(d4)

da1

dt

(c5) da1:=X;

(d5)

da1

dt

(c6) diff(a2,t);

(d6)

da2

dt

(c7) da2:=Y;

(d7)

da2

dt

(c8) 1/1/2+1/2da1^2+1/2x12da2^2+1/2x12da1^2da2^2-v;

(c9)

$$-v + \frac{da2^2}{2dt} + \frac{da1^2}{2dt} + \frac{da1^2}{2dt}$$

(c9) diff(diff(l,da1),t)-diff(l,a1)=0;

(c10)

$$2 \frac{da1}{dt} \frac{da2}{dt} + 12 \frac{da1^2}{dt} \frac{da2^2}{dt} + 11 \frac{da1^2}{dt} = 0$$

(c10) solve(X,diff(da1,t)); 58

$$(d10) \quad \left[\frac{d^2 a1}{dt^2} - \frac{12}{dt} \frac{da1}{dt} - \frac{da1^2}{dt^2} - \frac{da2^2}{dt^2} \right] = 0$$

(c11) diff(diff(l,da2),t)-diff(l,a2)=0;

$$(d11) \quad \frac{12}{dt} \frac{da2}{dt} - \frac{12}{dt} \left(\frac{da1}{dt} \right)^2 - \frac{da2^2}{dt^2} = 0$$

(c12) solve(Z,diff(da2,t));

$$(d12) \quad \left[\frac{da2}{dt} = \left(\frac{da1}{dt} \right)^2 - \frac{da2^2}{dt^2} \right]$$

(c13) p1=i1*da1+i2*da1*a2^2;

$$(d13) \quad p1 = i2 \frac{da1}{dt} a2^2 + i1 \frac{da1}{dt}$$

(c14) solve(Z,da1);

$$(d14) \quad \left[\frac{da1}{dt} = \frac{p1}{12 a2^2 + i1} \right]$$

(c15) p2=i2*da2;

$$(d15) \quad p2 = i2 \frac{da2}{dt}$$

(c16) solve(Z,da2);

$$(d16) \quad \left[\frac{da2}{dt} = \frac{p2}{i2} \right]$$

(c17) part(d13,2);

$$(d17) \quad \frac{12}{dt} \frac{da1}{dt} a2^2 + i1 \frac{da1}{dt}$$

(c18) p1:Z;

$$\frac{da1}{dt} a2^2 + \frac{da1}{dt}$$

(c18)

$$i2 \frac{dq1}{dt} a2^2 + i1 \frac{dq1}{dt}$$

(c19) cont(d15,2);

(c19)

ORIGINAL PAGE IS
OF POOR QUALITY

$$i2 \frac{da2}{dt}$$

(c20) #2:Z;

(c20)

$$i2 \frac{da2}{dt}$$

(c21) a:da1*a1+da2*a2-1;

$$(c21) \quad v + \frac{i2 \frac{da2}{dt} a2^2}{2} + \frac{da1}{dt} \frac{da1}{dt} a2^2 + i1 \frac{da1}{dt} \frac{da1}{dt} a2^2$$

(c22) returnend(X);

$$(c22) \quad v + \frac{i2 \frac{da2}{dt} a2^2}{2} + \frac{i2 \frac{da1}{dt} a2^2}{2} + \frac{i1 \frac{da1}{dt} a2^2}{2}$$

(c23) d14;

(c23)

$$da1 \quad a1$$

$$\left[\frac{da1}{dt} a2^2 + i1 \right]$$

(c24) subet(da1,a1/(i1+i2*a2*a2)*a2;c22);

$$(c24) \quad v + \frac{i2 \frac{da2}{dt} a2^2}{2} + \frac{i2 \frac{da1}{dt} a2^2}{2} + \frac{i1 \frac{da1}{dt} a2^2}{2}$$

(c25) diff(h,a1);

$$h = \frac{p_1^2}{2(I+I2 \phi i2^2)} + \frac{p_2^2}{2I2} + V$$

ORIGINAL PAGE IS
OF POOR QUALITY

```

(d27)
(c28) dp1=%;
(d28)
(c29) diff(h,a2);
(d29)
          da1 2
          i2 (---) a2
          dt
(c30) dp2=%;
(d30)
          da1 2
          i2 (---) a2
          dt
(c31) dp2%(-1);
(d31)
          da1 2
          - i2 (---) a2
          dt
(c32) dp2=%;
(d32)
          da1 2
          - i2 (---) a2
          dt
(c33) depends([w],[a1,a2,alpha1,alpha2]);
(d33)
          [w(a1, a2, alpha1, alpha2)]
(c34) diff(w,a1);
(d34)
          dw
          ---
          da1
(c35) p10=%;
(d35)
          dw
          ---
          da1
(c36) diff(w,a2);
(d36)
          dw
          ---
          da2
(c37) p20=%;
(d37)
          dw
          ---
          da2

```

da2

(c38) alpha2:=10^2;

(d38)

dw 2
(---)
da1

(c39) alpha1:=alpha2/(2*(1+a2^2))+10^2/2+v;

dw 2 dw 2
(---) (---)
da2 da1

(d39)

alpha1 = $\frac{da2^2}{2} + \frac{da1^2}{2(a2^2 + 1)} + v$

(c40) solve(% ,P20);

(d40)
$$\frac{dw}{da2} = \frac{sart(-(\frac{dw}{da1})^2 - 2a2^2v - 2v + 2alpha1a2^2 + 2alpha1)}{sart(a2^2 + 1)}$$

$$\frac{dw}{da2} = \frac{sart(-(\frac{dw}{da1})^2 - 2a2^2v - 2v + 2alpha1a2^2 + 2alpha1)}{sart(a2^2 + 1)}$$

(c41) part(d40,2);

(d41)
$$\frac{dw}{da2} = \frac{sart(-(\frac{dw}{da1})^2 - 2a2^2v - 2v + 2alpha1a2^2 + 2alpha1)}{sart(a2^2 + 1)}$$

(c42) depends([f],[alpha1,alpha2]);

(d42)
$$[f(alpha1, (\frac{dw}{da1})^2)]$$

(c43) part(d41,2);

(d43)
$$\frac{dw}{da2} = \frac{sart(-(\frac{dw}{da1})^2 - 2a2^2v - 2v + 2alpha1a2^2 + 2alpha1)}{sart(a2^2 + 1)}$$

ORIGINAL PAGE IS
OF POOR QUALITY

(c41) 'integrate(Z,a2);

$$(d44) \quad \int \frac{\frac{dw}{da1}^2}{\sqrt{-\left(\frac{dw}{da1}\right)^2 - 2a2^2 v - 2v + 2\alpha_{h1} a2^2 + 2\alpha_{h1}}} da2$$

$$\sqrt{a2^2 + 1}$$

(c47) w=alpha2^(1/2)*a1+d44+d42+k;

$$(d47) \quad w = \int \frac{\frac{dw}{da1}^2}{\sqrt{-\left(\frac{dw}{da1}\right)^2 - 2a2^2 v - 2v + 2\alpha_{h1} a2^2 + 2\alpha_{h1}}} da2$$

$$\sqrt{a2^2 + 1}$$

$$+ a1 \frac{dw}{da1} + k + f(\alpha_{h1}, \frac{dw}{da1})$$

(c48) a=2*(alpha1-v);

(d48) a = 2 (alpha1 - v)

(c49) c=alpha2/(2*(alpha1-v));

$$(d49) \quad c = \frac{\frac{dw}{da1}^2}{2 (\alpha_{h1} - v)}$$

(c51) 'integrate(1/(1-c/(1+a2^2))^(1/2),a2);

$$(d51) \quad \int \frac{1}{\sqrt{1 - \frac{c}{a2^2 + 1}}} da2$$

```
(c52) diff(w,alpha1)=1/s^(1/2)*%;
```

$$\frac{1}{\sqrt{1 - \frac{c}{a^2 + 1}}} da2$$

```
(d52)
```

$$\frac{dw}{d\alpha1} = \frac{1}{\sqrt{a}}$$

```
(c53) sort(d52,2);
```

$$\frac{1}{\sqrt{1 - \frac{c}{a^2 + 1}}} da2$$

```
(d53)
```

$$\sqrt{a}$$

```
(c2) 'integrate(1/(1-c/(1+a2^2))^(1/2),a2;0;a20);
```

$$\int_0^{a20} \frac{1}{\sqrt{1 - \frac{c}{a^2 + 1}}} da2$$

```
(d2)
```

```
(c3) beto1=(1/s^(1/2))*%;
```

$$\int_0^{a20} \frac{1}{\sqrt{1 - \frac{c}{a^2 + 1}}} da2$$

```
(d3)
```

$$\text{beto1} = \sqrt{a}$$

ORIGINAL PAGE IS
OF POOR QUALITY

(c4) integrate(1/((1+a2^2)*(1-c/(1+a2^2))^(1/2)),a2,0,a20);

$$(d4) \quad \int_0^{a20} \frac{1}{(a2^2 + 1) \sqrt{1 - \frac{c}{a2^2 + 1}}} da2$$

(c5) beta2=(1/(2*alpha1)^(1/2))*(a10+(alpha1^(1/2)/a^(1/2))*%);

ORIGINAL PAGE IS
OF POOR QUALITY

$$\text{sqrt}(\alpha 1) \int_0^{a20} \frac{1}{(a2^2 + 1) \sqrt{1 - \frac{c}{a2^2 + 1}}} da2$$

$$(d5) \quad \text{beta2} = \frac{\sqrt{a} \sqrt{2} \sqrt{\alpha 1}}{\sqrt{2} \sqrt{\alpha 1}} + a10$$

(c6) beta2=(1/(2*alpha1^(1/2)))*(a10+(alpha1^(1/2)/a^(1/2))*d4);

$$\text{sqrt}(\alpha 1) \int_0^{a20} \frac{1}{(a2^2 + 1) \sqrt{1 - \frac{c}{a2^2 + 1}}} da2 + a10$$

$$(d6) \quad \text{beta2} = \frac{\sqrt{a} \sqrt{2} \sqrt{\alpha 1}}{2 \sqrt{\alpha 1}}$$

(c7) d2;

$$(d7) \quad \int_0^{a20} \frac{1}{\sqrt{1 - \frac{c}{a2^2 + 1}}} da2$$

(c8) a1=beta2+alpha2*%;

$$(c9) \quad a1 = \alpha h a2 \frac{1}{\sqrt{(a2^2 + 1) \sqrt{1 - \frac{c}{a2^2 + 1}}}}$$

(c9) ds;

$$\frac{\sqrt{a20} \sqrt{\alpha h a1} \frac{1}{(a2^2 + 1) \sqrt{1 - \frac{c}{a2^2 + 1}}}}{\sqrt{a20} \sqrt{\alpha h a1} \frac{1}{(a2^2 + 1) \sqrt{1 - \frac{c}{a2^2 + 1}}}} + a10$$

$$(d9) \quad \beta a2 = \frac{\sqrt{a20} \sqrt{\alpha h a1} \frac{1}{(a2^2 + 1) \sqrt{1 - \frac{c}{a2^2 + 1}}}}{2 \sqrt{\alpha h a1}}$$

$$(c10) \quad \beta a2 = dw / d\alpha h a2;$$

$$(d10) \quad \beta a2 = \frac{dw}{d\alpha h a2}$$

$$(c11) \quad \text{'integrate'}((1/2)*(1/(a-b/(1+a2^2)))^(1/2)*((2*\alpha h a2)/(1+a2^2)),a2);$$

$$(d11) \quad \alpha h a2 \frac{1}{(a2^2 + 1) \sqrt{a - \frac{b}{a2^2 + 1}}}$$

$$(c12) \quad -Z*(-1);$$

$$(d12) \quad -\alpha h a2 \frac{1}{(a2^2 + 1) \sqrt{a - \frac{b}{a2^2 + 1}}}$$

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

$a_2^2 + 1$

(c13) beta2=a1+%;

(d13)

beta2 = a1 - alpha2

$$\frac{1}{(a_2^2 + 1) \sqrt{s - \frac{b}{a_2^2 + 1}}} \quad da_2$$

(c14) solve(%,a1);

(d14) [a1 = alpha2

$$\frac{1}{a_2^2 \sqrt{s - \frac{b}{a_2^2 + 1}} + \sqrt{s - \frac{b}{a_2^2 + 1}}} \quad da_2 + beta2]$$

(c15) closefile();

References:

1. Goldstein, H., Classical Mechanics, Addison-Wesley, Reading, Massachusetts, 1951.
2. Singh, S.N. and Schy, A.A., Invertibility and Robust Nonlinear Control of Robotic Systems, presented at 23rd IEEE Conference on Decision and Control, Las Vegas, Nevada, December 12-14, 1984.
3. Singh, S.N. and Schy, A.A., Robust Torque Control of an Elastic Robotic Arm based on Invertibility and Feedback Stabilization, present at 24th IEEE Conference on Decision and Control, Fort Lauderdale, Florida, December 11-13, 1985.
4. Punjabi, A. and Divins, J., Kinematics of Robot Arm Using MACSYMA, presented at NASA HBCU Seminar, April 20-22, 1986.
5. Punjabi, A. and Bivins, J., Solutions of Hamilton-Jacobi Equation for Robot Arm Using Symbolic Manipulation Program, presented at Virginia Academy of Sciences Annual Meeting, May 16, 1986.

FIGURES:

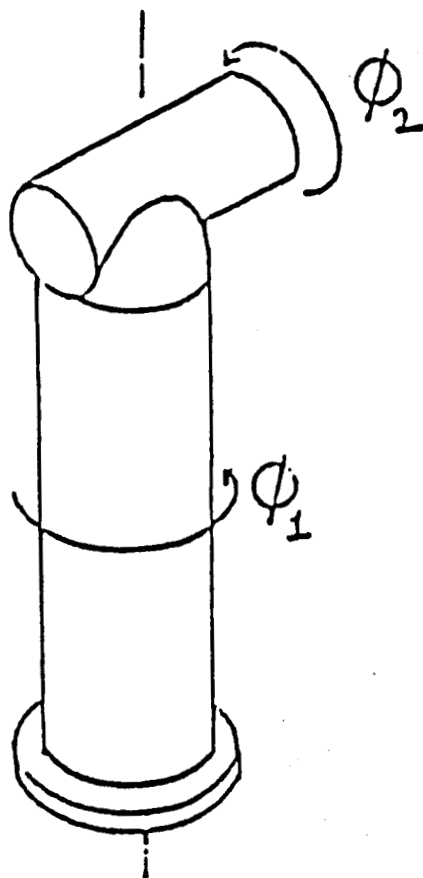


Fig. 1. Single link perpendicular robot arm. ϕ_1 is the angle of rotation of first arm and ϕ_2 is the angle of rotation of second arm.

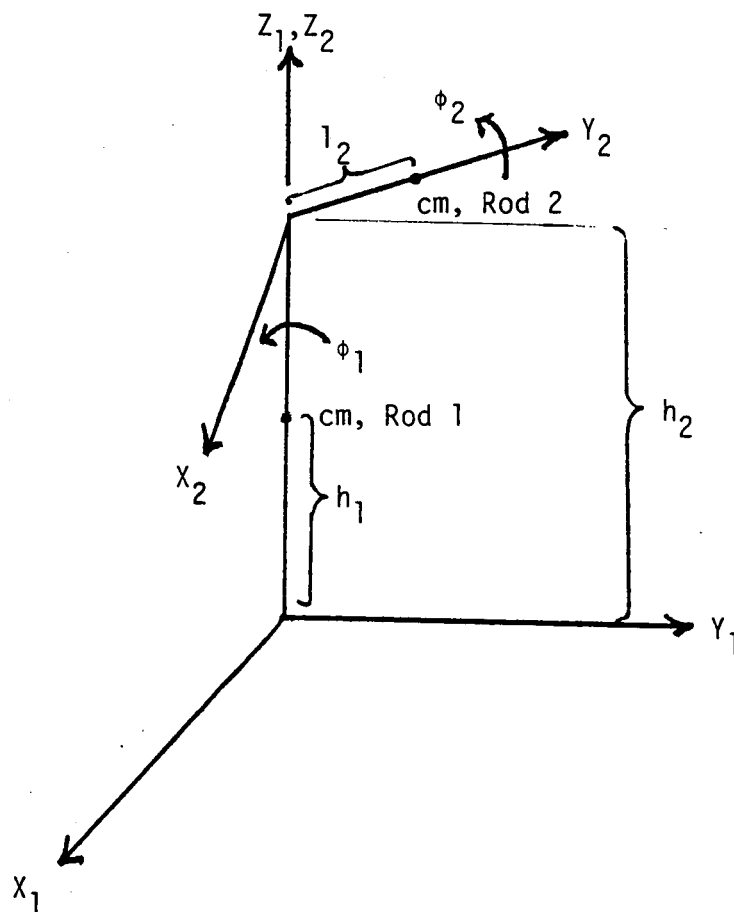


Fig. 2. Fixed and rotating co-ordinate systems $X_1Y_1Z_1$ and $X_2Y_2Z_2$ and other details used in analytic treatment.

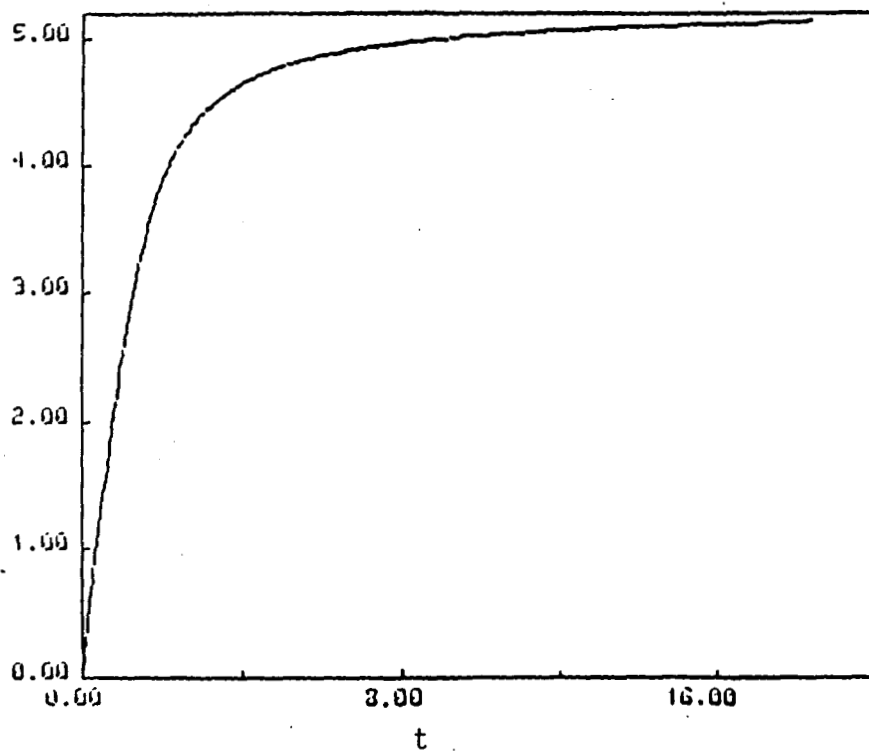


Fig. 3. Analytical solution of H-J equation with initial conditions: $\phi_1(0) = 0$, $\phi_2(0) = 0$, $p_1(0) = 1$, $p_2(0) = 1$, and $V = 1$. y axis represents ϕ_2 (in dimensionless form) as a function of time, and x axis represents time in seconds.

V. CONTRIBUTED PAPERS

The following papers were presented on the research conducted under this project.

1. Solutions of Van Der Pol's Equations Using MACSYMA
A. Punjabi and J. Bivins
NASA HBCU Graduate Student Workshop
NASA LaRC, Hampton, VA, March 1985
2. Kinematics of Robot Arm
A. Punjabi and J. Bivins
NASA HBCU Forum, Atlanta, Georgia, April 1986
3. Solutions of Hamilton-Jacobi Equation for Robot Arm Using Symbolic Manipulation Program
A. Punjabi and J. Bivins
Annual Meeting of Virginia Academy of Sciences, Harrisonburg, VA, May 1986
Virg. J. Sc., 37, 53 (1986)

Abstracts of these papers are shown on following pages.

KINEMATICS OF ROBOT ARM USING MACSYMA

A. Punjabi and J. Bivins

Department of Mathematics

Hampton University

Hampton, Va 23668

ABSTRACT

The Hamiltonian of the robot arm shown in the accompanying figure is given by:

$$H = \frac{p_1^2}{2(I_1 + I_2 \sin^2 \theta)} + \frac{p_2^2}{2I_2} + V$$

We obtain the Hamiltonian and attempt to solve the Hamilton-Jacobi equation for the characteristic function F_2 . We also attempt to get the Poincare surface of section for this problem. All this is done using one of the Symbolic Manipulation Programs. Our final goal is the study of nonlinear dynamical phenomena arising in this system.

References

1. Invertibility and Nonlinear Control of Robotic Systems, Singh, S.N. and Schy, A.A., 23rd IEEE Confec. on Decision and Control, Dec. 1984.
2. Regular and Stochastic Motion, Lichtenberg, A.J. and Liseberman, M.A., Applied Mathematical Sciences Series 38, Springer-Verlag, 1983.

SOLUTION OF THE HAMILTON-JACOBI EQUATION FOR SINGLE LINK ROBOT ARM USING
SYMBOLIC MANIPULATION LANGUAGES

Alkesh Punjabi and James Bivins

Department of Mathematics

Hampton University

Hampton, VA 23668

First the Lagrangian L for single link robot arm is set up in the fixed system of coordinates. Then the Hamiltonian H and the Hamilton's equation of motion are obtained. Hamilton-Jacobi equation for the characteristic function W is set up. H-J equation is solved in closed form. All these steps are done using the symbolic manipulation languages MACSYMA and SMP. (Supported by NASA under MACSYMA project.)